FP6- 506811

AMI
AUGMENTED MULTI–PARTY INTERACTION
http://www.amiproject.org/

Integrated Project
Information Society Technologies

D.5.2 IMPLEMENTATION AND EVALUATION RESULTS

Due date: 31/06/2006 Submission date: 15/08/2006

Project start date:1/1/2004 Duration: 36 months

Lead Contractor: DFKIGmbH 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)	

D.5.2 Implementation and Evaluation Results

Abstract:

Contents

1 Introduction

1.1 AMI and Multi-modal Multi-party Interaction

AMI is a multi-disciplinary 15-member consortium dedicated to the research and development of technology that will augment communications between individuals and groups of people. Partners in the AMI project conduct research focused on the use of advanced signal processing, machine learning models and social interaction dynamics to improve human-to-human communications, particularly during business meetings between co-located and remote (virtual) participants.

Workpackage 5 (WP5) of the AMI project encompasses all aspects of structuring and extracting information from multimodal meeting recordings. This work is based on the results from workpackage 4 (WP4) which is mainly concerned with the automatic recognition from audio, video and combined audio-video streams, with an emphasis on developing models and algorithms to combine modalities. More specifically, the initial problems targeted by WP4 concern robust speech recognition for multi-party meetings, gesture and action recognition, emotion recognition, source localization and object tracking, and person identification.

1.2 Content Abstraction

This document presents our efforts and achievements in content abstraction, indexing and retrieval, and summarization as carried out within workpackage 5 (WP5). This includes segmentation and classification tasks on many aspects such as Dialogue Acts, Chunking, Named Entity Recognition, Topic Segmentation, Addressing, Argumentation, and Hot Spot Detection as well as abstractive and extractive Summarization. For all tasks we have agreed on evaluation measures and procedures that allow us to compare different approaches and techniques as well as a comparison of our results with other work. In many areas, though, AMI is defining the field by extending basic approaches, e.g. from text and dialogue theory, to the interactive, multi-party setting and including multi-modal data.

Machine learning algorithms are at the core of most systems. In particular, our work on dialogue act recognition which also has been applied to a number of different corpora, compares different ML algorithms and we have extensive results from experiments with various feature sets.

Initial work in most areas had been done on the ICSI meeting corpus. Although this corpus provides only audio signals, it is a large and well-annotated corpus with realistic data. Also, with ICSI being a partner in AMI, we were able to get immediate and early access to some annotations, e.g. prosodic markup. In all areas of AMI research we are now looking at the AMI corpus data, usually the AMI hub scenario meetings. Such a common corpus is highly important as it allows us to design more complex systems as, for example, abstractive summarization, that draw on multiple sources of content abstraction.

Look-ahead Different applications can tolerate different amounts of processing lag, which affects how far ahead it is acceptable for the features to look. With off-line applications, it is even possible to use the entire meeting. Some on-line applications require immediate responses, and with others 10 second lags are fine. Our strategy is to have each task group agree on a look-ahead amount and describe the sorts of applications for which that amount is tolerable, thereby explaining what their technology is fit to do.

1.3 Fully Automatic Systems

Many of our systems rely on the transcription layer as the main input feature. Initial experiments have been conducted with the gold-standard manual transcription annotation in the AMI corpus. This also applies to other features that are currently only available as hand-coded annotations. We are currently preparing experiments to run the systems on the actual output of AMI's speech recognition (ASR) system and other recognisers. All infrastructure, such as data formats, scripts etc., is in place and we will repeat the current experiments with ASR

output and other automatically derived input as soon as it becomes available for the entire AMI hub scenario corpus.

We will publish a revised version of this deliverable that will include the results of the experiments with actual ASR output and other automatically generated features and their discussion. This version is planned for the end of November 2006.

In addition to the automatic system, some tasks will report two other types of results:

(1) results for the same kinds of features, but that use reference/gold standard versions of the features (reference transcription, hand-coded dialogue acts, etc.). This will allow us to judge the degradation we have to expect when applying our systems in an on-line setting, running directly from automatically generated transcripts and other features. Initial experiments, e.g. in Dialogue Act recognition, indicate that degradation of the quality of our systems roughly corresponds to the degradation in transcription quality. Note that using multiple (and multi-modal) input sources beyond the transcript makes the systems more robust and thus implicitly can actually correct ASR errors;

(2) results making use of non-automatic features for which we have no automatic processes (or where it would be too expensive to do the automatic version), showing whether or not adding the feature to what we can get automatically provides enough of an improvement that it would be worth attempting automatic processing in future.

1.4 Future Work

In the final months of AMI, we will work towards fully automated systems as outlined above. Most systems will be improved further, running extended experiments with various ML algorithms and feature sets. Work in abstractive summarization is ongoing, with more of the necessary annotations (e.g. ontology-based propositional content) and automated systems (e.g. semantic parsing) becoming available.

We will also finish currently ongoing work in a number of areas, including

- influence classification
- keyword spotting and indexing
- classification of meeting activities using conversational state sequences
- decision point detection
- sentence compression in extractive summarization

More details of our future plans are described in the final section of this deliverable.

1.5 Contributors

All AMI partners involved in workpackage 5 contributed to this deliverable: Brno University of Technology, DFKI, ICSI, IDIAP, TNO, University of Edinburgh, University of Twente and University of Sheffield. The main authors are: Jan Alexandersson, Rieks op den Akker, Jan Baan, Tilman Becker, Jan Černocký, Michal Fapšo, Alfred Dielmann, Yoshi Gotoh, Dirk Heylen, Sabrina Hsueh, Peter Huisman, Natasa Jovanovic, Thomas Kleinbauer, Wessel Kraaij, Stephan Lesch, Iain McCowan, Johanna Moore, Gabriel Murray, Barbara Peskin, Stephan Raaijmakers, Dennis Reidsma, Steve Renals, Rutger Rienks, Igor Szöke, Andy Thean, Jeroen van Rest, Daan Verbree, Alessandro Vinciarelli and Weiqun Xu.

2 Dialogue Acts

2.1 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 categorize 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 modelling the structure of a meeting. A simple example would be a browser that highlights all points where a suggestion or offer was recognized. Often, however, DA labels serve as elementary units to recognize higher levels of structure in a discourse. DAs may also control the processing of discourse content. To generate abstractive summaries, for example, content is extracted from utterances, and integrated in a discourse memory depending on the DAs of the utterances.

2.1.1 The Dialogue Act Recognition Task

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 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 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).

An integrated system may be used as a segmenter by ignoring its classifications. For purposes of comparison, it may also be used as a classifier, by forcing a human DA segmentation onto it; however, since the systems for the joint task are not optimized for the classification task, these results should be considered a baseline only.

2.1.2 Features

The features used in segmentation and classification are typically of the following types:

A language model based on words or part-of-speech tags. For a realistic evaluation of a segmenter or classifier, the words should be speech recognizer output; however, some of the results in this chapter are based on the human transcription, since large amounts of recognized speech from the AMI corpus are not commonly available yet.

Prosodic features like pitch and pitch slopes, the duration of words, vowels and pauses, and features derived from spectral coefficients.

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 feature.

Two important aspects to feature generation are the source and scope of the features. Eventually, all information required to generate features must come from automatic systems; however, information from annotations may be used to train systems. Also, systems are sometimes evaluated using features based on annotations, either because data from an automatic system is not available yet, or to assess the potential usefulness of a new type of feature. The

scope of features depends on the application that a system will be part of. If a system runs during the meeting, only past discourse is available. In a post-processing application, the whole discourse is available, allowing features which look forward from the currently processed utterance.

2.1.3 Metrics and Evaluation

The results of DA classification (using a given segmentation) are 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. Measuring the performance of segmentation or joint segmentation and classification is still an open topic. The systems described in this chapter are evaluated using a range of metrics, which are strict or more lenient, and refer to different units — words, boundaries, or dialogue acts.

It is important to note that all evaluations presented in this chapter are intrinsic, i. e., they are purely based on comparison between the human annotation and the classifier output; the effects of misclassifications depend on how the output is used, therefore they are not examined here. Ideally, the users of a DA segmenter/classifier provide information about the effects of different classification mistakes. Given such knowledge, an appropriate metric can be chosen and a DA classifier can be optimized for the specific consumer.

2.1.4 The AMI Dialogue Act Tag Set

The AMI scenario data consist of 35 sets of usually four meetings, with a length of roughly 72 hours. Most of the scenario data, over 100,000 utterances, have been annotated for dialogue acts. The AMI dialogue act scheme [1] consists of 15 dialogue acts (table 1), which are organized 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

2.1.5 The ICSI Meeting Corpus and DA Tag Set

The experiments reported in this chapter use the ICSI Meeting Corpus [Janin et al., 2003]. This corpus consists of 75 multiparty meetings recorded with multiple microphones: one head-mounted microphone per participant and four tabletop 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.

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-categorizations for a single DA unit. This extremely rich annotation scheme results in more than a

[1] 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

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%
All			102198	100.0%

Table 1: The AMI Dialogue act scheme, and the DA distribution in the annotated scenario meetings.

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., 2005, Zimmermann et al., 2006b]. 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 2. Note that statements are the most frequently occurring unit, 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 units.

Category	% of total DA units	% 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 2: Distribution of DA categories by % of the total number of DA units and by % of corpus length.

The corpus has been subdivided into three data sets: training set (51 meetings), development set (11 meetings) and test set (11 meetings). All our experiments were conducted on the same dataset subdivision proposed by [Ang et al., 2005] in order to have directly comparable results.

2.1.6 Related Work

[Stolcke et al., 2000] provide a good introduction to dialogue act modelling in conversational telephone speech, a domain with some similarities to multiparty 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 recognizers 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., 2005] 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., 2006b] 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., 2005], 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., 2005] 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.

2.1.7 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. Sections 2.2 and 2.3 describe two systems for the joint task with results on the ICSI corpus and preliminary results on AMI data. In sections 2.4 and 2.5, different machine learning approaches for segmentation and classification are compared on the ICSI data. Section 2.6 describes a classification system built for the AMI data. Finally, section 2.7 examines a method to select a reduced set of lexical features, with results on the ICSI and Switchboard corpora and preliminary results on AMI data.

2.2 DBN Based Joint Dialogue Acts Recognition

This section describes a joint segmentation and classification approach, using trainable statistical models: dynamic Bayesian networks (DBNs). We note that the full DA recogniser can be forced to operate on pre-segmented data, hence acting as a simpler DA tagger. Alternatively, by discarding the DA tags the system may be employed for the segmentation task alone.

We are interested in a DA dictionary composed of a few generic DA categories [Ang et al., 2005]. Classes of dialogue act in this scheme, which was obtained from the richer Meeting Recorder Dialogue Act (MRDA) annotation scheme [Shriberg et al., 2004], consisted of *statements*, *questions*, *fillers*, *back-channel* and *disruptions*. Those broad DA categories can be seen as the basic building blocks of a conversation, and thus they may be employed in modelling more complex meeting behaviours, such as meeting phases, or to enhance processes such as language modelling for automatic speech recognition or topic detection. These experiments were conducted on the

ICSI Meeting Corpus [Janin et al., 2003], using the same dataset subdivision (51 meetings for training purposes, 11 for development and 11 for testing) proposed by [Ang et al., 2005] in order to have directly comparable results.

Further experiments were conducted on the AMI Meeting Corpus using a richer DA annotation scheme and thus recognising 15 DA classes like: *backchannel*, *stall*, *fragment*, *inform*, *elicit inform*, *suggest*, *offer*, *elicit offer or suggestion*, *assess*, *elicit assessment*, *comment about understanding*, *elicit comment understanding*, *be positive*, *be negative* and *other*. The AMI corpus has been subdivided into two data sets: training set (95 meetings) and test set (23 meetings).

2.2.1 Methodology

Our framework for the integrated DA recogniser uses a generative approach composed of four main blocks: a Factored Language Model (FLM, section 2.2.3), a feature extraction component (section 2.2.2), a trigram discourse model, and a Dynamic Bayesian Network (section 2.2.4). The FLM is used to map sequences of words into DA units, and is the main component of the tagger. The discourse model consists of a standard trigram language model over DA label sequences[2]. Note that our DA tagger uses only lexical information and a discourse model. Experiments using both the reference orthographic transcription and the output of automatic speech recognition (ASR) have been carried out. The automatic transcription was provided by the AMIASR team and generated through an ASR system similar to the one outlined in [Hain et al., 2005b] (word error rate of about 29%). A set of six continuous features are used for DA segmentation purposes, together with part of a DBN model. This graphical model also plays a crucial role in the tagging process and acts as the master control unit for the entire recognition process.

2.2.2 Features

A vector of six continuous word related features was extracted from audio recordings and orthographic transcriptions.

Mean and variance of F0 Fundamental frequency (F0) was estimated using the ESPS pitch tracking algorithm get_f0[3] and sampled every 10 msec. The word temporal boundaries provided by the transcription [4] were then used to estimate the mean and variance of F0 for each word. Mean F0 was subsequently normalised against the speaker average pitch in order to have a participant independent feature.

RMS energy Average root mean square energy was estimated for each word W_i and then normalised by both the average channel energy (in order to compensate for factors such as channel gain and microphone position) and the mean energy for all tokens of word W_i.

Word length This is the word duration normalised by the mean duration for that word computed on the entire dataset. Therefore the resulting entity is inversely proportional to the rate of speech, neglecting estimation errors.

Word relevance The word relevance was computed to be the ratio between local term frequency within the current document and absolute term frequency across the whole meetings collection. Terms which are more relevant for the current meeting will assume scores well above the unity.

Pause duration Interword pauses were estimated using word boundary times obtained from aligning the transcription with the acoustic signal, and re-scaled in order to have a unitary range. Note that long pauses between words may highlight sentence boundaries and thus be a strong cue to DA segmentation. In fact pause related features have already been successfully employed in several DA segmentation frameworks (section 2.1.6).

[2]Estimated using the SRILM toolkit, available from http://www.speech.sri.com/projects/srilm/

[3]Available from http://www.speech.kth.se/snack/

[4]Note that word boundaries are estimated automatically through forced alignment between acoustic models and orthographic transcriptions, thus are characterised by a relevant amount of uncertainty.

2.2.3 Factored Language Models

Factored Language Models (FLMs) [Kirchhoff et al., 2002] are a generalisation of class-based language models in which words and word-related features are bundled together. The factors in an FLM may include word-related features such as part of speech, relative position in the sentence, stem, and morphological class. Indeed, there is no limit to the number of possible factors. In the FLM perspective even the words themselves, are usually considered one of the factors. Class based language models may be interpreted as a 2-factor FLM, in which words are bundled with classes.

Given a word f_t^0 and $k-1$ features $f_t^1, f_t^2, \ldots, f_t^{k-1}$, a sentence can be seen as sequence of these factor vectors $v_t \equiv \{f_t^0, f_t^1, \ldots, f_t^k\}$. As for standard language models, the goal of FLMs is to factorise the joint distribution $p(v_1, v_2, \ldots, v_n)$ as a chain product of conditional probabilities in the form $p(v_t \mid v_{t-1}, ..., v_{t-n})$. Since words have been replaced by vectors of factors, each conditional probability is now a function of these factors: $p(f_t^0, f_t^1, \ldots, f_t^k \mid f_{t-1}^0, f_{t-1}^1, \ldots, f_{t-1}^k, f_{t-2}^0, \ldots, f_{t-2}^k, \ldots, f_{t-n}^{0:k})$.

In order to build a good FLM it is necessary to choose the optimal factorisation (analogous to the structure learning problem in graphical models) and a backoff strategy to cope with data sparsity. Note that backoff is usually operated by dropping one or more factors from a Conditional Probability Table (CPT) in favour of a simpler conditional distribution (and smaller CPT), reiterating this procedure several times. Often multiple backoff paths (strategies) are feasible and it is even possible to concurrently follow all of them by adopting a generalized parallel backoff [Bilmes and Kirchhoff, 2003].

In order to model the relationship between words and DAs we have adopted a FLM based on three factors: words, DAs and the position of each word in the DA unit. Each word w_t is part of a DA unit and is characterised by the DA label d_t. Moreover each DA segment has been subdivided in blocks of five words: if w_t is one of the first five words the position factor n_t will be equal to one, if w_t belongs to the second block $n_t = 2$, and so on. The adopted language model is defined by a product of conditional probabilities $p(w_t \mid w_{t-1}, n_t, d_t)$. Note that considering only the word factor w_t the proposed FLM could be compared to a bigram since only the relation between w_t and w_{t-1} is taken into account. When backoff is required the first term to be dropped is the previous word w_{t-1}, leading to the backoff model $p(w_t \mid n_t, d_t)$. If a further backoff is required, the DA tag d_t will be dropped resulting in the simpler model: $p(w_t \mid n_t)$. We use Kneser-Ney discounting to smooth both the backoff steps.

In order to compare different FLM candidates, instead of comparing their perplexities, we have defined a simplified *DA tagging* task. We compare FLMs by measuring their ability to assign the correct DA label to unseen DA units. This preliminary evaluation was conduced by enhancing the FLM section of the SRILM toolkit [Stolcke, 2002] with a simple decoder, able to label each DA unit (sentence) with the most likely DA tag (factor label from a list of possible options).

The above described FLM, after training on the 51 meeting ICSI training set, was able to perform DA labeling on the 11 ICSI development set meetings with an accuracy of 69.7% using reference transcriptions and 63.4% using automatic transcriprions (70.9% and 63.6% on the 11 meetings from the test set). Replacing for example the word position factor n_t with part-of-speech tags p_t (automatically labeled by using a POS tagger trained on Broadcast News data) the accuracy on manual transcriptions fell to 61.7% (63.5% on the test set). Building the model $p(w_t \mid w_{t-1}, m_t, d_t)$, where m_t represents the information about the meeting type, the recognition rate rose to 68.2% (68.8% on the test set). A model including each of n_t, p_t and m_t with three backoff steps had slightly lower recognition rates of 67.7% on the development set and 68.2% on the test set.

2.2.4 Generative DBN Model

Bayesian Networks (BNs) are examples of directed acyclic Graphical Models (GMs). GMs represent a unifying concept in which probability theory is encapsulated inside the formalism of graph theory. Random variables are associated to nodes, and statistical independence between two random variables is represented by the lack of a connecting arc between the corresponding nodes. To model time series or data sequences, the BN formalism has been generalised into the Dynamic Bayesian Network (DBN) concept. A DBN is a collection of BNs where a single BN, with private intra-frame relations among variables, is instantiated for each temporal frame, and a set of

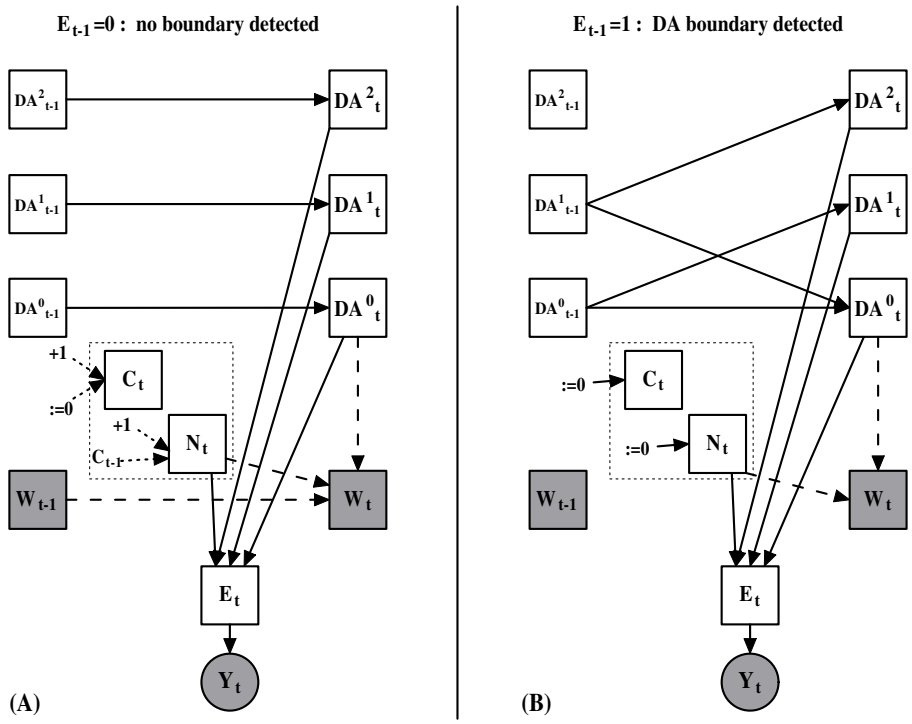

Figure 1: Overview of the DBN model for the integrated Dialogue Act recogniser. The model's topology depends on the state of the boundary detector E_{t-1} during the previous frame: the model's graph within a DA segment has been depicted on the left(A). The right side of the picture (B) shows the new topology immediately after a DA boundary detection. Shaded square nodes represent observable discrete variables, unshaded squares correspond to hidden discrete variables, and shaded circles are associated with continuous observations. Dotted arcs are not really part of the DBN: they symbolise relationships implied by the FLM.

inter-frame arcs is defined. Those connections between nodes of adjacent BNs explicitly describe the flow of time and help highlighting the temporal structure of each time-series.

A DBN is a modular and intuitive representation which provides a common underlying formalism [Murphy, 2002] for models including Kalman filters, Hidden Markov Models, coupled HMMs and hierarchical HMMs among the others. Note that since the DBN formalism is dual to a well defined mathematical theory, a unique set of tools and techniques can be developed to perform inference, model learning and decoding of any DBN model. The Graphical Model ToolKit (GMTK) [Bilmes and Zweig, 2002], for example, provides a formal language to describe DBNs and a common set of tools to experiment with them. Thus this toolkit has been adopted as the main development package for all the DBN related experiments described in this work. As anticipated in section 2.2.1 the DA recognition process is coordinated by a generative DBN based model. The overall model is depicted in figure 1. The node Y_t represents the continuous observable feature vector outlined in section 2.2.2 (associated to the word W_t). E is a binary variable that switches from zero to one when a DA boundary is detected. Since the node W_t represents a word, a DA unit can be interpreted as a sequence of words $W_{t-k},\ldots,W_{t-2},W_{t-1},W_t$ with a DA label DA^0 ($DA^0_{t-j} = DA^0, \forall j \in [0,k]$). DA^1 will contain the label of the previous DA unit, and DA^2 will go one more step back on the DA recognition history. C is a cyclical counter (from 0 to 5 and back to 0,1,2,...) which is used to count blocks of five words, and N accumulates the encountered word-blocks. Note that since the model's topology changes according to the state of the switching variable E_{t-1}, this is an example of a Bayesian multi-net [Bilmes, 2000].

Figure 1(A) shows the model's topology when a DA boundary has not been detected (intra-segment phase: $E_{t-1} = 0$). The current DA label DA^0 is responsible for the current sentence $W_t, W_{t-1}, \ldots, W_{t-k}$ and the joint sentence probability is estimated through the FLM $p(W_t \mid W_{t-1}, N_t, DA_t^0)$ introduced in section 2.2.3. Note that FLMs are fully supported by GMTK, which will automatically take care of the backoff procedure whenever required. The word block counter N needed by the FLM is automatically incremented whenever the cyclical word counter C reach the fifth word (word block dimension defined in section 2.2.3). All the DA label related nodes DA_t^k are simply copied from the previous temporal slice ($DA_t^k = DA_{t-1}^k$ with $k = 0, 1, 2$) since a new DA segment has not yet been recognised.

The state of the end boundary detector E is directly related to the word block counter N and the DA label history DA_t^k through a discrete CPT which is learned during training. The two states of E are linked to continuous feature vectors Y by two sets of Gaussian Mixture Models. Nodes E and Y (together with the associated CPT and GMMs) are fully responsible for the DA segmentation process. If the DA boundaries are known a priori, they can be injected into the model by making E an observable node, and the resulting system will operate as a DA tagger.

If during the previous frame $t-1$ a DA boundary has been detected, the model will be switched to the topology shown in figure 1(B) (inter-segment phase: $E_{t-1} = 1$). Since a new DA unit has been detected at the end of the previous frame $t-1$, both the counters C and N will be set to zero, and the FLM is forced to restart with a new set of estimations. The DA recognition history is updated by copying DA_{t-1}^1 into DA_t^2 and DA_{t-1}^0 into DA_t^1. The new DA hypotheses will be generated by taking in account the current DA label DA_{t-1}^0 and the previous one DA_{t-1}^1 through a trigram language model $p(DA_t^0 \mid DA_{t-1}^0, DA_{t-1}^1)$ (section 2.2.1).

The graphs in figure 1 show only the BN slices that are actually duplicated for $t > 1$. During $t = 0$ all the hidden states are properly initialised and the FLM is forced to backoff to $p(W_0 \mid N_0, DA_0^0)$ since W_0 is the first word. During the second frame $t = 1$, DA_1^2 is set to zero and the discourse language model is eventually forced to backoff to a bigram.

2.2.5 Experimental Setup and Performance Measures

All the experiments reported in this section have been performed on the ICSI corpus using the five DA categories outlined in section 2.2. The system outlined in the previous sections is primarily targeted on the DA recognition task intended as joint segmentation and classification, but as explained in section 2.2.4, it is possible to provide the ground truth segmentation and evaluate the DA tagger alone.

The percentage of correctly labeled units is about 76% on reference transcriptions and about 66% on ASR output. The classification procedure is exclusively based on the lexical information (through the FLM) and on the DA language model; prosodic related features are used only for segmentation purposes. Comparing these results with those shown in section 2.2.3, we can deduce that the introduction of a trigram discourse model has resulted in an absolute improvement included between 2% (on automatic transcriptions) and 5% (on manual transcriptions).

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 performances metrics proposed by [Ang et al., 2005] and subsequently extended by [Zimmermann et al., 2006b], together with a new recognition metric inherited from the speech research community. A detailed description of these metrics (NIST "Sentence like Unit" (SU) derived metrics, strict, lenient and boundary based metrics) can be found in [Ang et al., 2005]. The DA Error Rate (DER) and DA Segmentation Error Rate (DSER) are discussed in [Zimmermann et al., 2006b].

The speech recognition inspired metric derives from Word Error Rate but having words replaced by DA units. Recognised DA segments are firstly time-aligned against the ground truth annotation, and then the sum of substitution, deletion and insertions errors is scored against the number of reference DA units. This error metric is estimated using the publicly available tool SCLITE (part of the NIST Speech Recognition Scoring Toolkit [5]) which also provides detailed statistics on erroneous segments and significance tests. The SCLITE metric, compared with

[5] SCTK available from http://www.nist.gov/speech/tools/

all the other recognition metrics (except the lenient one), is more focused on a correct DA classification rather than on an extremely accurate segmentation.

Table 3 shows the segmentation and recognition results on five different setups. Results are reported using all the evaluation metrics cited above. Note that all the nine adopted metrics are "error rates", thus lower numbers correspond to better performances. The proposed setups differ only in the information used to detect DA boundaries: the *Lexical* setup makes no use of continuous features (node *Y* has been removed from the DBN), the *Prosody* setup uses only five out of six features (excluding pauses), the *Pause* setup uses the pause information but not the other continuous features, the *All (REF)* and *All (ASR)* configurations exploit the full feature set. *All (REF)* reports the results achieved by training and evaluating the DA recogniser on manually annotated orthographic transcriptions, whenever in *All (ASR)* the system has been developed and tested on automatic transcriptions. Therefore in the later experiment the combination of ASR and DA recogniser constitutes a fully automatic approach, since manual annotations are not needed. Note that the *Lexical* setup makes use of the lexical information just for DA

	Metric	Lexical	Prosody	Pause	ALL (REF)	ALL (ASR)
S	NIST-SU	93.7	83.4	48.0	**35.6**	**43.6**
E	DSER	83.6	90.7	51.2	48.9	58.2
G	STRICT	87.4	85.8	66.4	56.5	63.5
M.	BOUNDARY	14.5	12.9	7.4	5.5	7.3
R	SCLITE	52.7	60.7	48.8	44.6	53.5
E	NIST-SU	104.1	93.8	68.5	56.8	69.6
C	DER	86.7	92.1	62.9	61.4	72.1
O	STRICT	89.1	87.6	72.5	64.7	**72.5**
G.	LENIENT	20.7	22.0	19.5	**19.7**	**22.0**

Table 3: DA Segmentation and recognition error rates (%) of five different system configurations tested on the ICSI meeting corpus.

classification purposes. Boundary detection is estimated from the current DA label, the DA history and the word block counter. Therefore this setup and the lexically based systems investigated in [Zimmermann et al., 2006b] cannot be directly compared.

The adoption of prosodic and word related features made in the *Prosody* setup presents a conflicting behaviour: NIST-SU, strict and boundary metrics show an improvement over the baseline setup; while DSER, DER, lenient and SCLITE based metrics move toward higher error rates. The *Pause* setup shows a clear improvement over the baseline approach under all the evaluation metrics, and proves its strength over the *Prosody* setup highlighting the importance of pause related information on the segmentation task.

The fully integrated approach (*All-REF*) is the most accurate model. The error rates are similar to the NIST-SU segmentation error rate (34.4%) and the lenient recognition error rate (19.6%) of the two step recogniser presented by [Ang et al., 2005] (section 2.1.6). This result suggests that, even if the two competing systems have similar segmentation performances, and the maximum entropy based DA classifier (about 80% correct classification [Ang et al., 2005]) seems to be more powerful than our generative approach, the joint segmenter+classifier framework is potentially able to outperform a sequential framework. This is even more evident with the fully automatic ASR based system (*All-ASR*) which provides a relevant improvement if compared to the sequential approach outlined in [Ang et al., 2005] (lenient recognition error rate of 25.1%). In the sequential approach the DA classifier will be able to process only one segmentation hypothesis, whereas in the joint approach multiple segmentation hypotheses are taken in account by the DA tagger. The final choice between multiple candidates will be carried out by taking the most likely sequence of DA units, intended as the optimal combination of DA boundaries and DA labels.

2.2.6 Preliminary Experiments on the AMI Meeting Corpus

The same experimental setup adopted for the experiments carried out on the ICSI meeting corpus can be easily adapted to the AMI meeting data [Carletta et al., 2006]. FLM, features and DBN infrastructure can be applied to the new data virtually unaltered, the only difference is in the number of DA classes: 15 instead of 5 (section 2.2). The distribution of these 15 DA classes is quite imbalanced having more than 87% of the corpus concentrated in only 6 classes: *backchannel*, *stall*, *fragment*, *inform*, *suggest* and *assess*. Unfortunately DA tagging results are influenced by this distribution. The percentage of correctly classified units reach the 6.7% by drawing the DA classification by chance and 17.1% by taking also in account the prior distribution of the DA classes. If every units is classified as *inform* (the most frequent class) the percentage of correctly labeled units rises to a 34.5%.

Adopting the same experimental setup chosen for the FLM based DA tagging outlined in section 2.2.3, the percentage of correctly classified units is about 50.8% (well above chance results). Note that all the experiments reported in this section have been performed by using manually annotated orthographic transcriptions. Using the complete infrastructure depicted in figure 1 the tagging accuracy rises by an absolute 8% leading to 58.8%. Note that the 76% of correct classifications measured on ICSI data must be looked in perspective, since it was achieved on a simpler task (DA tagging dictionary 3 times smaller). The top of table 4 shows the segmentation

	Metric	AMI
S	NIST-SU	54.4
E	DSER	66.5
G	STRICT	74.1
M.	BOUNDARY	8.6
R	SCLITE	61.2
E	NIST-SU	86.4
C	DER	83.4
O	STRICT	87.2
G.	LENIENT	48.7

Table 4: DA Segmentation and recognition error rates (%) on the AMI meeting corpus.

error rates (for the full DBN based system) on the 23 testing meetings extracted from the AMI corpus. During the experiments on ICSI data, *pause duration* features had proved to be a relevant cue able to successfully highlight DA boundaries. Unfortunately *pause duration* features extracted from AMI data seem to be less reliable and lead to an higher percentage of missed DA boundaries.

The recognition error rates on the AMI corpus (bottom of table 4) are lower than the one measured on ICSI data (bottom of table 3). This is a direct consequence of the two previously outlined issues: the increase in the number of DA classes leads to a less precise tagging, and a more approximative segmentation implies that several DA boundaries are irreparably lost. Hopefully both these issues can be addressed effectively. DA tagging can be enhanced by investigating and tuning more discriminative FLMs. DA segmentation can be improved by reviewing the feature extraction process for the adopted features and by introducing new features.

2.2.7 Summary and Discussion

We have investigated the dialogue act recognition task in multiparty conversational speech, by applying a joint segmentation and tagging approach on natural meetings (AMI and ICSI meeting recordings). The proposed system makes use of a heterogeneous set of technologies: a graphical model, a factored language model and some continuous features. The graphical model, implemented as a DBN-based multi-net, oversees the whole recognition process. The proposed model adopts a generative paradigm for the DA tagging task and performs DA segmentation through a feature based architecture. DA tagging is performed using a factored language model over DA labels and word positions, together with a discourse language model. DA segmentation is obtained by exploiting both

the DA discourse model and a set of six continuous features extracted from audio recordings and orthographic transcriptions.

The joint DA recognition approach, if compared to a sequential one, provides a clearer view of the addressed problem and an intuitive strategy to its solution. The integrated approach encourages the reuse of common resources such as features and model parts. For example our graphical model shares the DA discourse model between the two subtasks (segmentation and classification), and makes the word block counter required by the FLM available for segmentation purposes (duration model). Furthermore the joint approach operates on a wider search space (producing joint sequences of segmentation boundaries and DA labels based on a trigram discourse model), and thus it is potentially capable of better recognition results. For example the results achieved in our reference transcription based experiments are similar to the sequential DA recognition approach proposed by [Ang et al., 2005], even though the maximum entropy DA classification approach chosen by the former work provides a 5% higher tagging accuracy. The advantage of a joint approach is substantial when manual orthographic transcriptions are replaced by imperfect automatic transcriptions. The lenient DA recognition error rate is degraded by only 2.3% and the comparison between sequential and joint approach is in favour of the latter one.

Moreover the initial experiments on the AMI meeting corpus can be seen as a baseline system and a starting point for further experiments. In the near future it is our intention to improve both DA classification and DA segmentation by enhancing the factored language model and by adopting a wider set of multimodal features.

2.3 Joint DA Recognition using HELM and Maxent Models

This section reports results for the joint segmentation and classification of multiparty meetings into its dialog acts (DAs). The text below concentrates on the joint system based on a combination of hidden-even LMs and maximum entropy models as described in [Zimmermann et al., 2006c].

2.3.1 Introduction

The task at hand consists in segmenting a stream of words into its individual DA segments and assigning correct DA types to the individual segments. This task description suggests a sequential solution where the stream of words is first split into individual segments that are then labeled in a separate step. In the case of joint segmentation and classification the task is performed in a single step.

The scheme for joint segmentation and classification proposed here is based on hidden-event language models (HELMs) and a maximum entropy (MaxEnt) classifier for the modeling of word boundary types. Specifically, the modeling of the boundary types takes dependencies between the duration of inter-word silence gaps and the word identities into account.

The earliest systems described in literature that performed both segmentation and classification of DAs was built in the framework of the VERBMOBIL project. A sequential approach was proposed in [Mast et al., 1996] while an A^* based joint segmentation and classification technique was investigated in [Warnke et al., 1997]. Based on the ICSI MRDA corpus [Shriberg et al., 2004], more recent work can be found in [Ang et al., 2005, Zimmermann et al., 2006b, Zimmermann et al., 2005, Zimmermann et al., 2006c, Dielmann and Renals, 2006]. In the case of [Ang et al., 2005] a sequential approach is adopted, the other references investigate joint segmentation and classification.

2.3.2 Method

The proposed approach for joint segmentation and classification of DAs is based on a combination of a hidden-event language model (HELM) [Stolcke and Shriberg, 1996] and a maximum entropy model.

The use of HELMs for the task at hand was investigated in [Zimmermann et al., 2006b]. The hidden events correspond to 6 different boundary types between consecutive words. The non-boundary, and 5 boundary types related to the DA types under consideration: Statements, questions, backchannels, floorgrabbers, and disruptions.

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 FA, 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 2: 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. 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 recognized elements are marked with a letter c while e is used to label errors.

The maximum entropy (MaxEnt) framework (see [Berger et al., 1996] for an excellent introduction) is used here to model both words and pause durations in a single framework similar to [Huang and Zweig, 2002]. This is different from the usual approach to model words and prosodic features by separate classifiers, where prosodic features (e.g. pause durations) are modeled by decision trees and word sequences by HELMs. As MaxEnt models (at least in the way they are typically supported by tool kits) assume all features to be in a binary form indicating either the presence or absence of a feature the pause durations have to be discretisized. For this durations from zero up to three seconds are partitioned into ten bins such that each bin received the same amount of samples. For pauses longer than three seconds an additional bin was used. In addition, up to 4 surrounding words are used as textual features. For the baseline case word features are omitted completely and p_i was the only feature used where p_i identifies the bin associated with the pause duration. Then, word w_i right before the pause and the joint feature (w_i, p_i) is included. Larger contexts contain more and more words to the left and to the right of the pause where not only the individual words are used as features but word bigrams as well. As in the case of HELM the MaxEnt model has to perform a 6-way classification. We therefor end up with probabilities for the same 6 event types.

For the final system HELMs and the MaxEnt classifier were combined as described in [Shriberg et al., 2000]. The probabilities provided by the MaxEnt model are weighted against those of the HELM with a log likelihood weight α. For $\alpha = 0$ the MaxEnt classifier does not influence the final result. Increasing α leads to a final result that is more and more influenced by the MaxEnt model.

2.3.3 Evaluation

All described methods are evaluated on the ICSI MRDA Corpus [Shriberg et al., 2004] based on both reference transcripts and the output of a speech-to-text (STT) system. The same data and the same split for training, development and test data is used as in [Ang et al., 2005].

Figure 2 illustrates a set of performance metrics for joint segmentation and classification of DAs. The NIST-SU error metric is a boundary based error metric introduced in [NIST website, 2003]. In contrast to the NIST error metric for segmentation (see Section 2.4.3) the type of the boundary is taken into account as well which leads

not only to false alarms and misses but substitutions as well. Both the strict and the lenient metric have been introduced in [Ang et al., 2005]. While the strict error metric requires correct DA boundaries the lenient metric completely ignores segmentation errors. In [Zimmermann et al., 2006b] the DA error rate (DER) was introduced. As the DER can also be defined via a DA based recall, DA based precision is defined here as well leading to a DA based F-measure: $F = 2 \times Recall \times Precision / (Recall + Precision)$.

2.3.4 Results

System	Reference	STT
Sequential [Ang et al., 2005]	64.4% (54.4%)	75.4% (64.3%)
Joint, no context	65.0% (55.9%)	76.2% (65.1%)
Joint, with word context	62.8% (51.0%)	73.6% (62.6%)

Table 5: Test set results for the systems investigated under both reference and speech-to-text (STT) conditions. The systems are the sequential approach proposed in [Ang et al., 2005], and the joint segmentation and classification system described here. The performance numbers represent the strict error metric and the DA recall (numbers in brackets).

Test set results for the investigated classifiers and experimental conditions (Reference, and STT) are reported in Table 5. Under reference conditions it is assumed that manual transcriptions are available while under STT conditions we have to rely on the output of a speech-to-text system. These results confirm the expected benefit of th use of word context for the modeling of the pause duration. A substantial improvement over the experiments that did not include word context is achieved.

2.3.5 Outlook

Future work should include the use of conditional random fields (CRF). CRF have an important advantage over MaxEnt models for joint segmentation and classification as they support the modeling of sequences and not only local features. In addition, further prosodic features based on pitch and energy could help to improve both segmentation and classification of DAs.

2.4 A Comparison of Systems for DA Segmentation

This section reports results for the segmentation of multiparty meetings into its dialog acts. The task at hand consists in a 2-way classification of inter-word boundaries into DA boundaries and non-boundaries. Below some of the segmentation systems developed in the framework of the DARPA GALE project based on decision trees (DT), hidden-event language models (HELMs) and maximum entropy (MaxEnt) are presented. Related material can be found in [Zimmermann et al., 2006a]. In addition, conditional random fields (CRF) are investigated here.

2.4.1 Introduction

DA segmentation is very closely linked to sentence segmentation that has been quite extensively studied over the past few years in order to enrich speech recognition output [Shriberg et al., 2000, Gotoh and Renals, 2000b, Huang and Zweig, 2002, Ang et al., 2005, Liu et al., 2005]. Combinations of HELMs and DTs were investigated in [Shriberg et al., 2000], the use of a MaxEnt classifier was studied in [Huang and Zweig, 2002], while [Liu et al., 2005] evaluated HELMs, MaxEnt, CRF. The performance for sentence boundary detection achieved in [Liu et al., 2005] could further be improved with a reranking technique [Roark et al., 2006]. Experiments are based on the ICSI MRDA corpus [Shriberg et al., 2004] under reference conditions (manual transcriptions) and speech-to-text (STT) conditions (automatic transcriptions).

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
Recall	.c.......c.....c.e...c
Precision	.c.e.e...c.....c...e.c

Metric	Counts	Reference	Rate
NIST-SU	3 FA, 1 miss	5 boundaries	80%
Recall	4 correct	5 boundaries	80%
Precision	4 correct	7 hypothesized boundaries	57%
F-Measure	-	-	67%

Table 6: The NIST-SU error rate, boundary based recall and precision metrics, and the corresponding F-Measure. 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 hypothesized boundaries are marked with a letter c while e is used to label false alarms and missed boundaries.

2.4.2 Method

Hidden-event language models (HELMs) for segmentation were introduced in [Stolcke and Shriberg, 1996]. They can be considered a variant of the widely used statistical *n*-gram language models [Jelinek, 1990]. The difference arises from the fact that during the training of the hidden-event language models the events to detect (sentence boundary tokens <s> in our case) are explicitly present, while they are missing (or hidden) during the recognition phase. For the combination of the HELM with decision trees and maximum entropy models covered below the integrated HMM scheme described in [Shriberg et al., 2000] is used.

Decision trees (DTs) based on the C4.5 algorithm [Quinlan, 1993] are used in combination with HELM for the baseline segmentation system. The decision trees are trained on the pause durations between two consecutive words that either correspond to a sentence boundary, or occur between two words of the same sentence. This is in contrast to other work, for example [Shriberg et al., 2000, Liu et al., 2005], where decision trees are trained on a large set of different prosodic features. The motivation for a pause-only system lies in the simplicity and low computational overhead of such an approach, as all the necessary information can be extracted from the ASR output alone.

Maximum entropy (MaxEnt) models [Berger et al., 1996] in their standard form assume all features to be in a binary form indicating either the presence or absence of a feature. In our experiments both words and pause durations are considered where the pause durations are binned into 10 classes. As features we use word and pause unigrams, word and pause bigrams, and bigrams of word and pause combinations according to [Zimmermann et al., 2006c][6].

The feature sets of the MaxEnt modeling are also used to train conditional random Fields (CRF) [Lafferty et al., 2001].

2.4.3 Evaluation

All classification methods are evaluated on the ICSI MRDA Corpus [Shriberg et al., 2004] based on both reference transcripts and the output of a speech-to-text (STT) system. The same split for training, development and test data is used as in [Ang et al., 2005]. See Section 2.5.3 for further details.

Table 6 illustrates the performance metrics used in the experiments described below. The NIST-SU error metric is a boundary based error metric introduced in [NIST website, 2003]. The familiar recall and precision numbers are are also boundary based. The F-measure is the harmonic mean of the computed precision and recall given the

[6]The features are computed using a 5-word window context (for the current, preceding two, and following two words).

reference sentence boundaries and the boundaries hypothesized by the segmentation system: $F = 2 \times Recall \times Precision/(Recall + Precision)$.

2.4.4 Results

System	Reference	STT Manual	STT Auto
HELM	46.4% (75.5%)	-	-
DT	57.0% (67.7%)	-	-
MaxEnt	37.1% (81.0%)	-	-
HELM+DT	36.5% (81.4%)	46.4% (76.0%)	53.4% (71.6%)
HELM+MaxEnt	34.8% (82.4%)	44.2% (76.5%)	50.7% (72.6%)
CRF	32.7% (83.1%)	40.8% (77.9%)	46.8% (74.8%)

Table 7: Test set results for the various classifiers and combinations investigated. The classifiers are hidden-event LMs (HELM), decision trees (DT), maximum entropy (MaxEnt), and conditional random fields (CRF). The performance numbers represent NIST-SU errors (and F-Measures in brackets).

Test set results for the investigated classifiers and experimental conditions (Reference, STT Manual, STT Auto) are reported in Table 7. The individual classifiers (with the exception of the CRF) are evaluated for comparison only using reference conditions (manual transcriptions). The results confirm the findings of [Ang et al., 2005] although the performance of the DT alone is lower than expected. The likely reason is the fact that we did not use bagging but rather, trained a single decision tree. As for joint segmentation and classification [Zimmermann et al., 2006c] the MaxEnt approach performs better than the HELM and the DT in isolation and the combination of HELM and MaxEnt consistently outperforms the HELM+DT combination under all experimental conditions. The CRF performs not only produces the single best results it also does better than the combinations of HELMs and DTs and MaxEnt models.

From the experiments with the CRF two further observations were made. First, the combination of the CRF with the HELM did not result in increased performance in contrast to the MaxEnt models and the DTs. Second, it appears that the reason for the better performance of the CRF cannot be attributed to the learning of (label) sequences but rather to the different training scheme. To support this hypothesis the CRF was not trained on sequences but rather on individual feature vectors to match the MaxEnt training setup. As the resulting performance was almost identical to the previous CRF experiments and still substantially better than for the MaxEnt models the only difference between the two approaches is to be found in the training algorithm. While the MaxEnt was trained using generalized iterative scaling (GIS) [Darroch and Ratcliff, 1972] the training of the CRF models was based on based on LBFGS [Liu and Nocedal, 1989], a quasi-newton algorithm for large scale numerical optimization problem. This observed difference in performance depending on the training algorithm for maximum entropy models is consistent with the findings of [Malouf, 2002].

2.4.5 Outlook

Future work should include boosting based segmentation according to [Zimmermann et al., 2006a] and support vector machines for classification, the use of further prosodic features and the integration of syntactic information such as automatically derived POS tags.

2.5 A Comparison of Systems for DA Classification

2.5.1 Introduction

This work compares the performance of various machine learning approaches and their combination for dialog act (DA) classification of meetings data assuming a correct segmentation of the data. For this task, boosting and three other text based approaches previously described in the literature are used. To further improve the classification performance, we also consider various combination schemes based on the results of the individual classifiers. Experiments are based on the ICSI MRDA corpus [Shriberg et al., 2004] under reference conditions (manual transcriptions) and speech-to-text (STT) conditions (automatic transcriptions).

Previous work mainly investigated single methods or variations of a single method for the classification of dialog acts. Most prominently, methods relying on word n-gram language models have been investigated in various experimental setups [Reithinger and Klesen, 1997, Nagata and Morimoto, 1994, Warnke et al., 1997, Stolcke et al., 2000]. Semantic classification trees have been proposed in [Mast et al., 1996], transformation based learning was investigated in [Samuel et al., 1998], and artificial neural networks were used in [Ries, 1999]. More recently, dynamic Bayesian networks have been proposed as well [Ji and Bilmes, 2005]. To our knowledge boosting based classification of dialog acts has not been investigated so far, and no direct comparison of the performance for the methods investigated in this work is available.

2.5.2 Method

All four DA classification methods described below attempt to predict the most likely DA type d^* for a given utterance $W = (w_1, w_2, ...w_n)$. In the paragraph below the most widely used technique, based on DA-specific language models, is covered first. Then a method relying on cue phrases and a maximum entropy based approach are described. Finally, the proposed boosting based method is introduced.

Dialog act-specific mini n-gram language models (Mini LMs) proposed by [Nagata and Morimoto, 1994] have been widely used in previous work [Reithinger and Klesen, 1997, Stolcke et al., 2000]. For each DA type d, an individual word n-gram LM is trained on all utterances from an annotated corpus that are tagged with the desired DA type d. This training procedure allows the Mini LMs to capture the DA specific word usage and produce DA specific likelihoods $p(W|d)$. To classify an unknown utterance W the estimates must then be multiplied by the prior probability $p(d)$ leading to the decision rule given below.

$$d = argmax_d p(W|d)p(d)$$

Although this method represents a principled approach that relies on the well known domain of n-gram language modeling it has the drawback of not being trained in a discriminative way.

The second technique investigated relies on the concept of cue phrases that correspond to word n-grams up to a specified order. The scheme has been proposed in [Webb et al., 2005] and is particularly simple to implement. During training the list of cue phrases is constructed in the following way. Initially cue phrase candidates include all word n-grams of a given corpus for $n = 1$ up to $n = 4$. For each such cue phrase C its predictivity $p(d|C)$ is computed that measures to which extend the presence of this cue phrase indicates the specific DA type d. For each cue phrase candidate its maximal predictivity that corresponds to the most likely DA type for the presence of this cue phrase is determined. Two thresholds are then used to obtain the final cue phrases. The first threshold requires a cue phrase candidate to be observed at least a given amount of times in the training corpus, and the second threshold only retains cue phrases that exceed a fixed minimal predictivity. For an unknown utterance W all known cue phrases are then extracted and the DA type corresponding to the cue phrase that is associated with the highest predictivity is used to output the result d. In our implementation we used the system including position specific cues by explicit modeling of the start and the end of utterances; see [Webb et al., 2005] for further details. A potential drawback of this method lies in its decision rule that does not generalize well to produce a score for each available DA type.

One of the main drawbacks of the methods described above lies in their training that does not explicitly optimize the discrimination between correct and incorrect DA types for a given utterance. To take advantage of discriminative training a DA classification technique based on maximum entropy modeling was proposed in [Ang et al., 2005]. Furthermore, the maximum entropy framework supports the direct estimation of posteriors $p(d|F)$ for a DA type d and a binary feature vector F. See [Berger et al., 1996] for an excellent introduction into maximum entropy modeling. The DA type d of an unknown utterance is determined by the DA type d that maximizes the posterior probability $p(d|F)$. In our case the feature vector F is extracted from the utterance W according to [Ang et al., 2005]. As features the first two words, the last two words, the initial and the final word bigram, as well as the length of the utterance is used. In contrast to [Ang et al., 2005] we do not include the first word of the following DA, as our experimental setup only considers isolated utterances.

The fourth method, based on boosting, is also discriminative and is derived from a text categorization task. Boosting aims to combine weak base classifiers to come up with a strong classifier. The learning algorithm is iterative, and in each iteration, a weak classifier is learned so as to minimize the training error, and a different distribution or weighting over the training examples is used to give more emphasis to examples that are often misclassified by the preceding weak classifiers. For this approach we use the BoosTexter algorithm described in [Schapire and Singer, 2000], with word n-gram features, as well as features like segment length. To make it comparable we have also trained a BoosTexter model with the same set of features as for the maximum entropy approach.

2.5.3 Evaluation

We use a tightly controlled experimental setup that only allows the methods to access isolated utterances (i.e. the classification is based on words within a given utterance and does not make use of other knowledge sources such as the sequence of utterances or prosody). All classification methods are evaluated on the ICSI MRDA Corpus based on both reference transcripts and the output of a speech-to-text (STT) system.

DA Type	Ref	STT Manual	STT Auto
Statements	8,918	8,642	7,740
Questions	1,164	1,108	1,004
Backchannels	1,960	1,437	220
Floor-grabbers	1,924	1,768	1,306
Disruptions	2,237	1,937	1,734

Table 8: Test set frequencies of the DA types under reference conditions (Ref), STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto).

The same split for training, development and test data is used as in [Ang et al., 2005]. Of the 75 available meetings in the ICSI MRDA corpus [Shriberg et al., 2004], two meetings of a different nature are excluded (Btr001 and Btr002). From the remaining meetings we use 51 for training, 11 for development, and 11 for evaluation. The available DA types are mapped to the following five mutually exclusive types: backchannels (B), disruptions (D), floor grabbers (F), questions (Q), and statements (S). See Table 8 for the test set frequencies of the five DA types.

In contrast to [Ang et al., 2005] we use the normalized words coming from forced alignments under reference conditions instead of the unnormalized words from the meeting transcriptions. For the speech-to-text (STT) conditions we also make use of a better, more recently developed recognizer [Stolcke et al., 2006]. Instead of a 39% word error rate (WER) the new recognizer achieves a 35.4% WER based on the close talking microphones. Furthermore, we define two separate STT conditions. The first one corresponds to the STT conditions of [Ang et al., 2005] and relies on a manual segmentation of the audio input stream (STT Manual). As a more realistic setup we also include an experimental setup that relies on automatic segmentation of the audio (STT Auto) leading to a higher WER of 38.2%.

2.5.4 Results

For the Mini LM approach described in [Nagata and Morimoto, 1994] the DA-specific n-gram LMs were trained and optimized up to n = 4 on the development set. Significantly different results (using a sign test) were obtained for the step from unigram LMs (error rate 36.7%) to bigram LMs (error rate 27.5%). Trigram and fourgram Mini LMs performed slightly worse but not significantly different from the bigram LMs.

For the Cue Phrase method described in [Webb et al., 2005] we measured the error rates for different maximum lengths of the cue phrases. In correspondence with [Webb et al., 2005] the best results were achieved when cue phrases up to 4-grams were used (error rate: 27.3%). In contrast to the Mini LM approach the higher order n-grams significantly helped to improve the performance over the use of bigrams only, for which an error rate of 30.4% was measured.

In the case of the maximum entropy based method described in [Ang et al., 2005], the effect of the number of words to include at the beginning and at the end of each dialog act was investigated. The number of the initial words and final words to include as features turns out not to be very critical for the performance of this method. When only the first and the last word is kept, an error rate of 22.9% is achieved under reference conditions compared to 22.6% for keeping the first two words (plus the initial word bigram) and the final two words (and the final word bigram).

With BoosTexter [Schapire and Singer, 2000], we have not performed a full optimization and ran the classifier for 1,000 iterations for each experimental condition. Using all unigrams, bigrams, and trigrams of an utterance as features results in a classification error rate of 22.1%. When the length of the utterance is included as a feature, the classification error is reduced to 21.7%. A similar error rate, 21.9% is achieved when the BoosTexter is trained on the same features as the maximum entropy based method. These results indicate that the words at the beginning and the end of an utterance carry most of the information that can be exploited by the classification scheme.

System		Ref	STT Manual	STT Auto
Mini LM	[Nagata and Morimoto, 1994]	26.7%	29.8%	27.3%
Cue Phrases	[Webb et al., 2005]	26.6%	29.6%	28.5%
MaxEnt	[Ang et al., 2005]	22.5%	26.5%	23.8%
BoosTexter		21.7%	26.9%	24.2%
Simple Voting		23.8%	27.3%	24.4%
Linear Combination		21.7%	26.3%	23.6%
MLP		21.3%	26.2%	23.3%

Table 9: Comparison of the classification error rates of the different systems under reference conditions (Ref) , STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto). The results for the combination schemes are at the bottom.

After the individual optimization of each DA classification method, the best performing configuration was used for evaluation on the test sets under the three available conditions. The resulting test set error rates are reported in Table 9. In correspondence with [Webb et al., 2005] we find that the performance of the approach based on cue phrases compares well with the mini LM based approach, in spite of the difference in both corpus and DA type definitions. The main result from Table 9 is the observation that the two approaches based on mini LM, and cue phrases perform significantly worse than the maximum entropy based approach and the classification scheme using boosting under all investigated conditions.

In a first experiment a simple voting scheme was implemented that returns the DA type most frequently predicted by the different classifiers (the most frequent DA class is chosen in case of ties). According to Table 9, voting did worse than either the maximum entropy approach (MaxEnt) or the boosting based method (BoosTexter). As a second combination method, linear interpolation of the posteriors of the DA classification methods was investigated. This combination method performed better than the simple voting scheme and under the STT conditions linear interpolation even outperformed the individual classifiers (at a 90% level of significance). Only

the last combination scheme based on a multilayer Perceptron (MLP) was able to significantly outperform (at the 99% level) the best individual classifiers under all conditions. For this combination method a simple feed-forward network with a single hidden layer including ten hidden neurons was trained on the development sets.

Specifically, our results indicate that both the boosting based approach and the method relying on maximum entropy significantly outperform the use of mini language models and the scheme relying on cue phrases. The best performance was achieved by a combination method that involved a multilayer perceptron.

2.5.5 Outlook

Future work could include support vector machines for classification, the use of prosodic features and the integration of syntactic information such as automatically derived POS tags.

2.6 DA Classification using Maximum Entropy Models

This section describes dialogue act classification on AMI data, based on the ground truth segmentation using maximum entropy (maxent) modelling. We employ a freely available maxent classifier toolkit by the Stanford NLP group. [7]

2.6.1 Maximum Entropy Models

We use a conditional model of the probability of a class *cl* (the DA type) given an observation *o* (features derived from an utterance):

$$P(cl|o) = \frac{e^{\sum_i \lambda_i f_i(o,cl)}}{\sum_{cl'} e^{\sum_i \lambda_i f_i(o,cl')}}$$

The observation o is a set of features describing the unit to classify. The f_i are indicator functions defined on o and cl. Each f_i is assigned a weight λ_i. The weights are the model parameters which are estimated from the training material, i. e., observations whose classes are known.

In the maxent toolkit used here, there is an f_i for each feature/class pair (f, cl), and λ_i indicates how strongly the presence of this feature suggests that the utterance in question is of the class cl, as illustrated in table 10. x

	bck	stl	fra	inf	el.inf	sug	off	el.sug	ass	und	el.ass	el.und	be.pos	be.neg	oth	Range
yeah	**2.98**	**2.05**	-0.12	1.34	-0.97	-0.31	-0.78	-1.98	**2.99**	0.68	-0.57	-2.20	-1.01	-1.92	-0.18	5.20
sorry	-0.51	-0.55	-0.50	-0.82	-0.09	-0.43	-0.42	-0.36	-0.83	**1.31**	-0.42	-0.34	**4.30**	0.13	-0.47	5.12
mm-hmm	**4.15**	-0.03	-0.31	0.96	-0.87	-0.58	-0.86	-0.86	**1.61**	0.69	-0.87	-0.86	-0.86	-0.86	-0.44	5.03
so	-1.25	**3.05**	**1.61**	0.89	0.21	0.15	-0.58	-0.60	0.35	-1.03	-0.04	-1.64	-0.46	-0.80	0.15	4.69
well	-1.07	**3.61**	0.60	0.69	0.10	0.19	-1.03	-1.01	**1.08**	-0.89	-0.49	-0.79	-0.09	-0.79	-0.13	4.68
mm	**3.24**	**1.26**	-0.01	-0.05	-1.40	-0.18	0.17	-1.22	**1.29**	0.18	-0.99	-1.08	-1.10	-1.08	0.97	4.64
thanks	-0.19	-0.24	-0.26	-0.44	-0.21	-0.27	-0.19	-0.19	-0.74	-0.19	-0.20	-0.19	**3.68**	-0.19	-0.19	4.43
okay	**1.89**	**1.64**	-1.76	-0.35	-1.30	-0.09	-0.70	-1.28	**2.28**	**2.04**	0.28	0.82	-0.94	-1.97	-0.57	4.25
thank	-0.16	-0.53	-0.43	-0.53	-0.44	-0.37	-0.14	-0.21	-0.48	-0.30	-0.25	-0.15	**3.69**	-0.11	0.41	4.22
show	-0.05	-0.67	-0.82	-1.29	-0.22	0.35	**2.78**	-0.23	-0.59	-0.09	0.20	-0.06	-0.19	-0.07	0.94	4.07
i'll	-0.28	-0.28	**1.22**	-0.48	-0.84	-0.75	**3.17**	-0.37	-0.52	-0.22	-0.60	-0.11	-0.40	0.46	0.00	4.01

Table 10: An example illustrating the meaning of the weights in a maxent model. Row headers are features (words), column headers are classes (DAs), each table entry is the weight of a feature/class pair. The rightmost column shows the weight range of the feature, which can be interpreted as its distinctiveness. The table lists the most distinctive features of a DA model trained on words only; the highest weights for each feature are printed in boldface.

[7] We would like to thank Christopher Manning and Dan Klein of the Stanford NLP group (`http://nlp.stanford.edu`) for providing the maxent classifier package for Java. The software is publicly available at `http://nlp.stanford.edu/software/classifier.html`

Since the maxent toolkit used is limited to binary features, it is most suitable for features which are binary by nature, like words and word n-grams in relatively small, sentence-like units. Numerical features, especially real-valued ones like the duration of an utterance, have to be represented as binary features. This is usually done by splitting the value range into intervals, and mapping each data value to one of the intervals. The definition of the interval boundaries is critical, and the conversion incurs a loss of information, because numerical order is lost. Also, since observations are represented by sets of features, it is easy to add or remove features. A potential drawback is that maxent models do not capture correlations between features: if the co-occurrence of two features has a particular meaning, it is up to the developer to generate an additional feature which represents the co-occurrence.

Introductions to maxent modelling can be found in [Berger et al., 1996, Berger, 1997, Ratnaparkhi, 1998, Klein and Manning, 2003].

2.6.2 Features

The following types of features were defined:

Lexical features All n-grams (n=1,2,3,4), specially marked utterance-initial and utterance-final n-grams and the whole text string of the utterance. All lexical features have been generated from reference transcriptions (word level).

Length and duration The length (number of words) of the utterance is used as a feature as-is; the duration is discretised by taking the integer of the logarithm. These features are defined for the current utterance, as well as the last utterance of the current and the previous other speaker.

Non-word features Laughs and disfluencies have been transcribed and are used as features. It is, however, not clear yet whether speech recognition will be able to provide these features.

Temporal relation Four features indicate the temporal relation between the current utterance i to the previous utterance $i-1$, in the order of start times.
Pause: $endTime_{i-1} + 0.1s < startTime_i$
No pause: $endTime_{i-1} - 0.1s < startTime_i \leq endTime_{i-1} + 0.1s$
Overlap: $startTime_i \leq endTime_{i-1} - 0.1s$
Containment: $endTime_i < endTimePrev_{i-i}$
Additionally, these relations may be indexed with the DA label of the preceding utterance, e.g. to indicate that the current utterance overlaps with a DA of type Backchannel.

Speaker change There are two versions of this feature: one is based on the speaker of the previous and current utterance, the other is triggered by a pause between the current and the previous utterance of the current speaker.

Dialogue act history The DA label of the immediately preceding utterance (as ordered by start time), the previous utterance by the current speaker and the last utterance by another speaker. To train models, these features were generated from the annotated DA labels; during evaluation, they were generated from the previous classifications.

These features were generated on the basis of a real-time assumption, that is, only the current and preceding utterances are used. For applications which post-process a meeting after it was held or allow for delayed processing, this restriction could be weakened or removed.

2.6.3 Method and Performance Measures

The experiments were performed on the 'scenario' meetings of the AMI corpus, which consists of 35 series of (usually) four meetings. The scenario data was split into a training set of 25 series, a development set of five series

and an evaluation set of five series (table 11). For cross-validation, the union of the training and evaluation sets was split into ten folds; the development set was omitted from cross-validation, since the selection of feature types was obtained using the development set. Since the ground truth segmentation was used, evaluation is a simple comparison against the human annotation. For each DA type, we define recall and precision, and for the whole set, we define accuracy (the mean recall) and the mean precision.

$$
\begin{aligned}
correct_{DA} &= \textit{the number of times DA was correctly classified} \\
annotated_{DA} &= \textit{the number of occurrences of DA in the annotated test data} \\
tagged_{DA} &= \textit{the number of times DA was classified} \\
Recall_{DA} &= \frac{correct_{DA}}{annotated_{DA}} \\
Precision_{DA} &= \frac{correct_{DA}}{tagged_{DA}} \\
Accuracy &= \frac{\sum_{DA} correct_{DA}}{\sum_{DA} annotated_{DA}} \\
Precision &= \frac{\sum_{DA} Precision_{DA} * annotated_{DA}}{\sum_{DA} annotated_{DA}}
\end{aligned}
$$

Subset	Meetings	#meetings	#DAs
Training set (Train)	ES2002, ES2005-2010, ES2012-2016 IS1000-1007 TS3005 TS3008-3012	98	69380
Development set (Dev)	ES2003, ES2011, IS1008, TS3004, TS3006	20	17082
Evaluation set (Eval)	ES2004, ES2014, IS1009, TS3003, TS3007	20	15736
All scenario data		138	102198

Table 11: The split of the AMI scenario data into training, development and evaluation sets. The column marked #DAs indicates the number of DA annotations in the subsets; not all meetings have been annotated for DAs.

2.6.4 Feature Selection

For efficiency reasons, features occurring less than three times in the training material were omitted. Since adding features does not necessarily improve the accuracy of the model, a simple round-based, iterative selection algorithm was applied to the feature types defined above:

1. Start with an empty set of selected feature types, and a set of candidate types
2. Add each feature type tentatively, train a model and test its performance on the development set
3. If any types lead to an improvement when added, permanently add the type which leads to the largest improvement, and go to step 2).

The result of the feature selection is a smaller set of feature types, which performs as well as all feature types while it contains only half as many binary features. From the lexical features, words, initial and final words, bigrams, four-grams and the whole utterance string were selected; simple words are the strongest cues. Also selected were the segment length, speaker change, temporal relation indexed with the DA of the preceding segment, and the previous DA type uttered by the current speaker.

2.6.5 Results

Models were trained using all features and the selected feature types, and evaluated on the development and evaluation sets. Furthermore, cross-validation was performed on the union of the training and evaluation sets, using the selected feature types. Table 12 compares the results. Using the selected feature types and testing on the evaluation set, we reach an accuracy of 65.8%. The frequencies, recall and precision of the single DAs for this experiment are shown in table 13. Table 14 is the confusion matrix. The most frequent confusions are bck/ass, und/ass, inf/ass, inf/sug, inf/fra and stl/fra. To some degree, these ambiguities are reflected in the weights of the words in the model excerpt shown in table 10 — some of the lexical cues are inherently ambiguous. For instance, the strongest cue, the word "yeah", predicts Backchannel and Assess with equal strength.

Features	Trained on	Evaluated on	Accuracy	Precision
All	Train	Dev	62.1%	61.5%
All	Train	Eval	65.7%	64.7%
All	Train+Dev	Eval	65.9%	65.0%
Selected	Train	Dev	62.3%	61.8%
Selected	Train	Eval	65.8%	64.9%
Selected	Train+Dev	Eval	65.7%	64.9%
Selected	Train+Eval (cross-validation)		63.6%	62.6%

Table 12: DA classification results. Results for the whole set of feature types and the selected types are similar, and there is no significant difference when the Dev set is included as training material. However, the Dev set appears to be easier than the Eval set.

As table 12 shows, accuracy and precision on the development and evaluation sets differ by three to four percent and the cross-validation result lies inbetween, which indicates that the evaluation set is 'easier' and the development set is 'harder' than average. This assumption is supported by the class distribution: the three classes with the highest recall (Fragment, Backchannel and Inform) cover 48.6% of the development set, 54.2% of the whole scenario corpus, and 57.3% of the evaluation set. Similarly, the majority class baseline obtained by tagging all utterances as Inform is 25.6% for the development set, 28.3% for the whole corpus and 31.5% for the evaluation set.

The accuracy of the maxent tagger is slightly better than the preliminary results obtained by running the DBN system described in section 2.2.6 on AMI data using the reference segmentation. However, the results are difficult to compare due to the different nature of the systems: while the maxent tagger relies on correctly pre-segmented utterances, the DBN system combines segmentation and classification, and the reference segmentation was provided to it. Future work will include evaluating the maxent tagger on automatic segmentation.

2.6.6 Outlook

The major areas for future work will be the input used to generate features, and the features themselves. The system will be adapted to use ASR transcriptions and automatic segmentation. Also, the usefulness of other signals like head and hand gestures will be examined. The existing feature types will be enhanced to include forward-looking features. Also, since the maxent framework is not aware of correlations between features, the generation of new features by analyzing correlations of existing features will be examined.

2.7 Dialogue Act Tagging using smart Feature Selection; Results on Multiple Corpora

2.7.1 Introduction

The topic of automatic Dialogue Act classification has received a fair amount of attention in the past years [Jurafsky et al., 1998, Rotaru, 2002, Shriberg et al., 2004] (see also Table 15). A variety of methods have been

DA	annotated		tagged		correct		Recall	Precision
bck	2025	12.9%	2197	14.0%	1584	15.3%	78.2%	72.1%
stl	943	6.0%	794	5.0%	539	5.2%	57.2%	67.9%
fra	2034	12.9%	2094	13.3%	1656	16.0%	81.4%	79.1%
inf	4962	31.5%	5578	35.4%	3854	37.2%	77.7%	69.1%
el.inf	563	3.6%	347	2.2%	212	2.0%	37.7%	61.1%
sug	1155	7.3%	1109	7.0%	520	5.0%	45.0%	46.9%
off	233	1.5%	134	0.9%	84	0.8%	36.1%	62.7%
el.sug	51	0.3%	38	0.2%	12	0.1%	23.5%	31.6%
ass	2670	17.0%	2919	18.5%	1589	15.4%	59.5%	54.4%
und	259	1.6%	69	0.4%	21	0.2%	8.1%	30.4%
el.ass	224	1.4%	110	0.7%	46	0.4%	20.5%	41.8%
el.und	16	0.1%	5	0.0%	5	0.0%	31.2%	100.0%
be.pos	314	2.0%	203	1.3%	170	1.6%	54.1%	83.7%
be.neg	9	0.1%	1	0.0%	0	0.0%	0.0%	0.0%
oth	278	1.8%	138	0.9%	59	0.6%	21.2%	42.8%
Total	15736		15736		10351		65.8%	64.9%

Table 13: Frequencies, recall and precision for the individual DAs in a model trained with the selected features and evaluated on the evaluation set. For example, the evaluation set contains 2025 backchannels (12.9%). 2197 utterances (14.0%) were tagged as backchannels, 1584 times correctly (15.3% of the correctly classified utterances). 1584 out of 2025 backchannels were recognized (recall 78.2%), and 1584 out of 2197 utterances classified as backchannels were actually backchannels (precision 72.1%). The overall accuracy is 65.8%.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	annot.
1	bck	**1584**	48	14	16	2	1	.	.	342	9	.	.	1	.	8	2025
2	stl	79	**539**	139	29	4	5	.	.	130	1	2	.	.	.	15	943
3	fra	20	111	**1656**	137	12	26	2	.	58	1	5	.	1	.	5	2034
4	inf	57	24	109	**3854**	44	365	31	4	435	1	11	.	16	1	10	4962
5	el.inf	3	3	36	193	**212**	37	5	11	33	3	20	.	2	.	5	563
6	sug	3	4	25	511	9	**520**	4	4	53	1	12	.	3	.	6	1155
7	off	6	1	7	90	5	15	**84**	.	21	1	1	.	.	.	2	233
8	el.sug	.	1	5	9	8	8	.	**12**	1	.	7	.	.	.	.	51
9	ass	362	28	58	489	8	95	4	.	**1589**	20	5	.	.	.	12	2670
10	und	52	11	3	33	5	2	.	.	118	**21**	.	.	7	.	7	259
11	el.ass	5	6	9	64	26	17	3	6	40	1	**46**	.	.	.	1	224
12	el.und	1	.	.	2	2	.	.	.	5	.	1	**5**	.	.	.	16
13	be.pos	.	.	11	68	2	11	.	1	40	4	.	.	**170**	.	7	314
14	be.neg	.	.	1	5	.	.	.	.	2	.	.	.	.	**.**	1	9
15	oth	25	18	21	78	8	7	1	.	52	6	.	.	3	.	**59**	278
tagged		2197	794	2094	5578	347	1109	134	38	2919	69	110	5	203	1	138	
correct		1584	539	1656	3854	212	520	84	12	1589	21	46	5	170	.	59	
incorrect		613	255	438	1724	135	589	50	26	1330	48	64	.	33	1	79	

Table 14: A confusion matrix for the classifications in table 13. Rows are the ground truth DAs, columns are classified DAs. The number in row y column x indicates how often an utterance which is annotated as DA y has been tagged as DA x by the classifier; numbers on the diagonal represent correct classifications. The column "annotated" and the rows "tagged" and "correct" are as defined in section 2.6.3. The most frequent confusions are bck/ass, und/ass, inf/ass, inf/sug, inf/fra and stl/fra.

tested on various corpora using different dialogue act classes. This can make the comparison between different methods rather difficult. It is well known that the words and phrases in DA's are the strongest cues to their identity [Jurafsky et al., 1998]. When looking at current state-of-the-art DA tagging, we may conclude that experiments that are easily and unambiguously replicable and that compare the performances on different corpora have not yet been conducted. This section describes the first session of a series of experiments that tries to adhere to these issues. We next describe the three corpora that we used, provide an overview of previous work on these corpora, explain our approach and then compare our performances on the three corpora with known results. Finally, we present first results on DA classification on ASR output, instead of on manual transcriptions.

2.7.2 Various Corpora

For our DA tagging experiments we have used three different corpora, each with their own tagset: the ICSI Meeting Corpus [Janin et al., 2003], the Switchboard Corpus [Godfrey et al., 1992] and part of the AMI corpus [Carletta et al., 2005]. We briefly describe each of these in turn.

The ICSI Meeting Corpus includes 75 naturally occurring meetings containing roughly 72 hours of multi-talk speech data and associated human generated word-level transcripts. It was hand-annotated for dialog acts as described in [Dhillon et al., 2004, Shriberg et al., 2004] using the Meeting Recorder Dialog Act tagset (MRDA). The MRDA scheme has 11 general tags and 39 specific tags. Each annotation requires one general tag and a variable number of specific tags. The MRDA scheme proposes several classmaps as well in which several tags are grouped together. For our experiments on the ICSI corpus we have used the widely applied classmap that maps the DA's onto 5 distinct classes: statements (S), questions (Q), backchannels (B), fillers (F) and disruptions (D). Utterances that have not been annotated are labelled (Z).

The ICSI Corpus comes along with a proposed train/test split. This split consists of 51 meetings (almost 80.000 utterances) which can be used for training, 11 meetings (about 13.500 utterances) for development, and 11 meetings (over 15.000 utterances) for evaluation. This split leaves out 2 of the 75 meetings. These are excluded because of their different nature.

The Switchboard Corpus is a corpus of conversational speech by telephone. For our experiments, we used the same subset of the corpus as [Stolcke et al., 2000]. The subset consists of over 210,000 utterances grouped in 1,155 conversations. Dialogue act annotations based on the SWBD-DAMSL tagset are available for all of these conversations [Jurafsky et al., 1997]. Similar to [Stolcke et al., 2000], we used the clustered tagset containing 42, out of the original 220 DA-labels.

The AMI Corpus is a collection of over 100 hours of four person project meetings. All meetings are in English. However a large proportion of speakers are non-native English speakers. Amongst a lot of signals, the transcriptions of all meetings are available as well as several *layers* of annotations. Since not all the dialog annotations of the meetings were available at the time we ran the experiments, we used a subset of 80 meetings[8]. Our collection comprises about 50.000 utterances. The AMI DA tagset has 15 tags : Backchannel, Stall, Fragment, Inform, Elicit Inform, Suggest, Offer, Elicit Offer Or Suggestion, Assess, Elicit Assessment, Be Positive, Be Negative, Comment About Understanding, Elicit Comment About Understanding, and Other.

2.7.3 Previous Work

Baseline A baseline used for comparing the accuracy of classification results, is the majority class baseline. We choose however, to compare the performance with the performances achieved using the set of manually acquired cue phrases, known as the LIT set, as proposed by Samuel [Samuel, 2000]. This set contains 687 different cue phrases that have been assembled from several papers, dissertations and books. Table 16 gives an overview of the baselines computed for the three corpora. For the Switchboard corpus we were unable to compute the LIT set performance on a machine with 2048 MB internal memory.

[8]These were: ES2002ACD; ES2003BCD; ES2006; ES2007; ES2008; ES2009; ES2010; ES2011ABC; ES2012CD; ES2014ABC; ES2015; ES2016; IS1000A; IS001; IS1003; IS1004; IS1005ABC; IS1006ABD; IS1008; IS1009; TS1003ABC; TS1004ABC; TS1005; TS1007A.

Feature / Article	[Ang et al., 2005]	[Rosset and Lamel, 2004]	[Fernandez and Picard, 2002]	[Rotaru, 2002]	[Lendvai et al., 2003]	[Andernach, 1996]	[Reithinger and Klesen, 1997]	[Venkataraman et al., 2002]	[Keizer and Akker, 2005]	[Venkataraman et al., 2005]	[Jurafsky et al., 1998]	[Zimmermann et al., 2006b]	[Zimmermann et al., 2005]	[Warnke et al., 1997]	[Katrenko, 2004]	[Webb et al., 2005]
Sentence length	X								X							
First two words	X	X														
Last two words	X															
First word of next segment	X															
Speaker		X														
Number of utterances		X														
Prosodic			X		X						X			X		
Bigrams of words in segment				X												
(Correct) last 10 DA's					X											
Words in last 10 DA's					X											
Utterance type						X										
Presence/absence 'Wh'-words						X										
Subject Type						X										
Specific cue phrases						X					X				X	
First verb type						X										
Second verb type						X										
Question mark						X										
Polygrams of words in segment							X							X		
Ngrams of words in segment								X		X		X	X			X
Ngrams of previous DA's								X		X			X			
Specific patterns									X							
Previous DA									X							
Next DA									X							
Grammar pattern									X		X					

Table 15: Features used for DA-classification in different studies

Corpus	**Majority Class**	**LIT set**
ICSI	55.46	63.57
Switchboard	33.73	uncomputable
AMI	28.69	44.24

Table 16: Baselines for the different corporas

Performances Most studies on dialogue act classification have used one of the corpora that we use in our experiments. We present the best results obtained, as far as we know, for the ICSI and Switchboard corpus. As the AMI Corpus is a new corpus no previous DA tagging results have been published yet.

The best performance for DA classification on the ICSI corpus currently is 81.18%, as reported in [Ang et al., 2005]. The classification obtained, was performed on the same train/test split and using almost (omitting 'Z' class) the same classmap as mentioned before in Section 2.7.2. This results in a classification task with 5 distinct classes.

Previous research on the DA classification of the Switchboard Corpus has been reported by Rotaru et al. [Rotaru, 2002] and Stolcke et al. [Stolcke et al., 2000]. Stolcke obtains a 71% accuracy using a trigram word model whereas Rotaru achieves 72%. Unfortunately both use different fixed train/test splits, without mentioning which split has been used for these results. Furthermore the description of the methods used are not very clear. Stolcke uses trigrams and Rotaru uses bigrams as one of the features, but in both cases it is unclear if n-gram selection methods have been used. Another difference between the two studies is that Rotaru included the utterances labelled with the '+' DA-tag, whereas Stolcke excluded these. Based on these descriptions it is not possible to reproduce nor to compare their results.

2.7.4 Our Approach

We tried to devise an experiment that was both replicable. Our used feature-set contained the following categories:

?/OR: Whenever a question mark is present, the number of times the word *or* appears is counted and used as a feature.[9]

Length: The length (number of words) of each segment.

Last Labels: A bi-gram of the previous two labels.

N-grams (compressed): At first, all bi-, tri- and quadri-grams of words were computed for all tagged utterances. Then we applied the n-gram selection method which selects the Top-N most predictive n-grams using conditional probability scores [Samuel et al., 1999] for both the Top-N of the uni, bi-, tri- and quadri-grams individually (order specific) and the Top-N of the merge of all n-grams (non order specific) for each class. The remaining n-grams are used for calculation of one single feature value for each speech act. This is done by awarding a number of points based on the match with the preselected n-grams (See [Verbree, 2006] for more details). Instead of a 'presence' value for each individual pre-selected n-gram, all information is now *compressed* into one feature value, saving a factor N of features, which enables e.g. performance computations on the huge Switchboard corpus.

POS-N-grams (compressed) The POS-N-gram features were computed in a similar fashion as described for the word n-grams. POS-tag features have been scarcely used for DA-classification, if at all. [Cathcart et al., 2003] have used them for back-channel classification, but none of the papers presented in Table 15 mentioned its usage.

The biggest difference with our approach in comparison to earlier approaches is the use of a *compressed* feature set for the N-gram and POS-N-gram features. This technique enabled us, in contrast to e.g. [Ang et al., 2005] to make use of each word of the utterance and unlike e.g. [Zimmermann et al., 2005] we did not end up with an extremely large feature set.

Table 17 shows the abbreviations used in the subsequent tables in which we present our classification results.

2.7.5 Results

All results presented were obtained by using the J48 classifier using the default settings as available in Weka [Witten and Frank, 2000]. J48 was chosen after a careful selection of classifiers in which both performance and computational time were taken into account. [Verbree, 2006]. For computation of all our results, 10-fold cross validation was used.

[9] Until now our classification is based on transcriptions in which the question mark is available, but eventually we aim to base our classification on ASR-output in which this feature might not be available anymore.

Abbreviation	Feature
L	Length
P	Uni-, bi-, tri- and quadrigrams of POS-tags
W	Uni-, bi-, tri- and quadrigrams of words
QMT	Question mark token
ORT	'OR' token
LL	Last Label
NOS	Non-Order Specific
OS	Order Specific
C	Compressed
I	Individual
T10	Top 10

Table 17: Features and their abbreviations

To compare the effect of the compression, the results of the compressed set are (when computationally possible) compared with the feature set containing the combination of all the individual (Top-N) Ngrams.

For the classification results on the ICSI corpus we used the train/test split provided. Contrary to Ang & Shriberg [Ang et al., 2005], we did not exclude utterances of the class 'Z'. The results obtained in our experiments are depicted in Table 18.

Feature set and parameters chosen	Performance
L_P_W (NOS) (C T10)	87.84
L_P_W (OS) (I T10)	87.97
QMT_ORT_L_P_W (OS) (C T10)	87.82
QMT_ORT_L_P_W (OS) (I T10)	87.98
QMT_ORT_L_LL_P_W (OS) (C T10)	89.13
QMT_ORT_L_LL_P_W (OS) (I T10)	89.27

Table 18: DA-classification results on the ICSI Meeting corpus

On the Swichboard corpus we used the 42 DA's from the classmap (similar to Stolcke et al.) and also included the *'+'* DA by using it on its own. (The *'%-'* DA was mapped on the *'%'* DA.) Note that this makes the results not *directly* comparable to those of Stolcke and Rotaru. To overcome this, we have also performed an experiment in which we discarded the utterances of the class *'+'*, resulting in an accuracy of 70.26%.

Our performances on the Switchboard corpus is shown in Table 19. Similar to the inability to compute the performance of the LIT set on the Swichboard corpus, we were unable to compute results for the classifiers using individual n-grams. This is mainly due to the amount of distinct DA tags used and the size of the corpus. It should be said that our computations on the Switchboard Corpus still required a huge amount of computing power, even for the compressed feature set. The process of part-of-speech tagging, ngramming and classifying all 10 folds created out of the 210,000 utterances available in the Switchboard Corpus took about 3 days for each classifier-setting.

For the AMI corpus, we were unable to make use of the *Last Label* feature as at the time of writing not all the annotated meetings contained dialog acts with timing information. This is a pity because the feature set now differs from the set used for the other corpora. Close inspection of the feature's contribution on the results for the other two corpora showed that its added value was in the order of one or two percent [Verbree, 2006]. Table 20 lists all the obtained performances. [10]

[10] At the time of writing, there was not decided on the exact training and test split, this will be the case for the final version

Feature set and parameters chosen	Performance
L_P_W (NOS) (C T10)	60.57
L_P_W (OS) (I T10)	uncomputable
QMT_ORT_L_P_W (OS) (C T10)	60.22
QMT_ORT_L_P_W (OS) (I T10)	uncomputable
QMT_ORT_L_LL_P_W (OS) (C T10)	65.68
QMT_ORT_L_LL_P_W (OS) (I T10)	uncomputable

Table 19: DA-classification results on the switchboard corpus

Feature set and parameters chosen	10 fold
QMT_ORT_L_P_W (OS) (C T10)	53.52
QMT_ORT_L_P_W (OS) (I T10)	57.82
L_P_W (OS) (I T10)	57.94
L_P_W (NOS) (C T10)	53.24

Table 20: DA-classification results on the AMI corpus

2.7.6 Results on ASR

In order to move one more step in the direction of fully automatic DA classification, we have performed a DA classification experiment on 11 AMI meetings[11] from which the ASR output was available. The DA-labels used originated from the annotations performed on the manual transcripts. The resulting corpus consisted of 8374 utterances. The experiments were run in a 10 fold cross-validation setup. Table 21 shows the performances obtained for this ASR corpus in comparison to the results using the manual transcriptions for the same set of meetings.

Feature set and parameters chosen	ASR	Manual
QMT_ORT_L_P_W (OS)(C T10)	37.43	53.74
QMT_ORT_L_P_W (OS) (I T10)	40.05	56.89
L_P_W (OS) (I T10)	40.26	56.55
L_P_W (NOS) (C T10)	37.05	51.29

Table 21: DA-classification results on Manual and ASR transcriptions

The decrease in performance, which was expected, can in potential be blamed to the word error rate of the speech recognizer. However, also the smaller corpus size may have played an important role, as some of the features that were used become more useful on a larger data-set (c.f. language models). We need to wait until more data is available before we can further address this issue.

2.7.7 Discussion

Closer analysis of the QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10) classifications on the ICSI corpus (see its confusion matrix in table 22) shows one of the most interesting challenges for future work. It appears that a lot of *backchannels* are misclassified as *statements*. Analyzing the ngrams selected which should cue for *backchannels* it appears that, even if an ngram is among the best-cueing ngrams for a specific class it might even cue more for another class. This phenomena was observed for all three corpora examined.

[11] ES2002ACD; ES2009ABCD; IS1009ABCD

a	b	c	d	e	f	$<--$ classified as
1558	1	18	0	384	0	a = B
55	1869	131	125	60	4	b = D
133	100	990	0	91	3	c = F
0	12	0	1095	4	4	d = Q
471	2	14	19	8057	6	e = S
5	4	3	0	3	177	f = Z

Table 22: Confusion matrix of QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10) ICSI setting

As a result of this, also reflected in Table 22, the more frequently occurring classes also have a larger chance of being classified correctly. Normalization in a preprocessing phase could potentially overcome this.

A result of our multi-corpus approach is that it brings us in a position where we are able to investigate the impact of various corpus structures on the performance. Readers should note that, since our classification results are better than Shribergs' and equivalent to Stolcke's, this does not e.g. legitimates to infer that Stolcke's classification performance on Switchboard outperforms Shriberg's performance on the ICSI corpus. One cannot say this because features that work well on one corpus could work even better, or worse on a different corpus.

2.7.8 Conclusions

In this section we presented a method of DA tagging using a *compressed* feature set that apart from using words also used the more general part-of-speech-level of a sentence. Results on different corpora show a major improvement over the majority class baseline as well as over the LIT set baseline. Furthermore our classification outperforms earlier results obtained on the ICSI set sets a inter-corpora standard for the Switchboard and AMI corpus, using a replicable 10 fold cross-validation approach. The results on the ASR output show a large decrease in performance.

3 Summarization

3.1 Introduction

This study concentrates on various methods of extractive and abstractive meeting summarization. While automatically generated *abstractive* summaries, similar to human-authored summaries, employ novel sentences to succinctly describe the content of the original document, *extractive* summarization aims to locate informative portions of the original document, remove those portions from their original contexts, and concatenate them together to form a "cut-and-paste" type summary. The advantages of extractive summarization over abstractive summarization are that no prior domain-specific knowledge is required, no language generation component is needed, and the extracted sentences or *dialogue acts* can easily serve as indices into the transcript or audio record. The advantage of abstractive summaries is their coherent text style even if the primary source for the summary, i. e., the utterances made by the meeting participants, is potentially ungrammatical and incoherent. In this respect, automatically generated abstractive summaries resemble more the kind of human-authored summary that is familiar to naive users.

For extractive summarization, a major focus of this study is to see whether speech-specific features such as prosody or speaker status supplement or even outperform purely lexical or textual features. Because text summarization is a more well-developed field with a depth of literature, it is sensible first to apply text summarization approaches to meeting transcripts and thus essentially treat a meeting record as a textual document. However, it is hypothesized that the differences between spontaneous speech data and written data are significant enough that summarization approaches incorporating speech-specific sources of information will outperform others.

Evaluating these various summarization approaches is itself a controversial and unsettled issue. There are intrinsic evaluations of automatic summaries, which compare the information content of a system summary to model human summaries, and there are extrinsic evaluations which investigate how well an automatic summary helps a real-world information retrieval task, for example. While intrinsic evaluations are easy to run and thus are attractive for the development stage of a summarization system, there are questions concerning how well these intrinsic metrics correlate with human evaluations. In the case ob abstractive summaries, the application of intrinsic methods is even more problematic. As the contents of the summary are produced by a language generation component, the resulting text might contain terms, such as synonms, that never appeared in the original meeting. Most intrinsic methods, however, rely on comparions of terms from the source document(s) and the summary text. It is apparent that such metrics can not do justice to the actual quality of an abstractive summary. We therefore decided to concentrate on an extrinsic evaluation method for abstractive summarization, as described below. For extractive summaries, this study runs multiple evaluations and compares the results, with an aim toward deciding on the best evaluation framework for the remainder of the project.

This research is motivated by the idea that extractive or abstracive summaries can be incorporated into a meeting browser framework, with which a user can access the transcript and audio records via the summary itself. There are multiple potential use cases for such a browser. One is that a user has missed a meeting and wants to understand the gist of the discussion and whether or not any important decisions were made, perhaps so that s/he can report the findings to a supervisor. Another use case is that a user actually attended a meeting but wants to quickly review the meeting as a refresher. In the latter case, the user might be able to simple rely on a keyword search to navigate the transcript, but in the former case the user would not even know what to search for if they didn't have any idea of what the meeting was concerning. It is for this type of scenario that we hypothesize an automatically generated summary would be very useful.

3.2 Research Motivation

Figure 1 provides a simple demonstration of how a summary might be incorporated into a meeting browser framework. The main browser window is a splitter window with two panels that can be arranged horizontally (stacked on top of each other, as shown) or vertically. In this figure, the top window contains the extractive summary, which is 700 words long and consists of the most important dialogue acts of the meeting. The user might first read the

summary in its entirety, then choose to navigate to portions of the transcript using the buttons next to each summary dialogue act. In this way, the summary dialgoue acts behave as indices into the meeting record. Once the user navigates to a portion of the meeting transcript which is interesting to them, they can choose to play the audio file beginning at that point of the meeting. This audio facility is particularly important, given the recognition errors which will exist in the speech recognition transcript. Meanwhile, the smaller window in the foreground with horizontal stripes provides the user with a global layout of the meeting's informativeness, with blue lines indicating the original positions of the extracted dialogue acts and the red line indicating their current location within the meeting transcript. The user can also use this smaller window as an alternative way of navigating through the meeting.

Although Figure 1 is only a simplified version of a meeting browser, it reinforces the point that extractive summaries are not meant to be stand-alone documents, but rather to serve as navigation tools. In fact, extractive summaries may sometimes be too fragmented or allusive to make complete sense to a user on their own. By making access to the transcript and audio records more efficient, extractive summaries can aid a user's information retrieval needs.

3.3 Extraction Experiments

The following sections detail a series of experiments in which numerous extraction techniques are implemented and compared with one another. The discussion of evaluation is presented throughout the discussion of these experiments and subsequently in a separate section of the section.

3.3.1 First Experiment: MMR, LSA, and Prosodic Features

The first experiment in this study compared two summarization approaches inherited from the field of text summarization, Maximal Marginal Relevance (MMR) and Latent Semantic Analysis (LSA), with a machine learning approach relying on prosodic features of pitch, energy and duration. This experiment is described in more detail in [Murray et al., 2005a]. It uses the ICSI Meeting Corpus ([Janin et al., 2003]) with a test set of 6 meetings.

MMR The first text summarization approach implemented was MMR. MMR ([Carbonell and Goldstein, 1998]) uses the vector-space model of text retrieval and is particularly applicable to query-based and multi-document summarization. The MMR algorithm chooses sentences via a weighted combination of query-relevance and redundancy scores, both derived using cosine similarity. The MMR score $Sc^{MMR}(i)$for a given sentence S_i in the document is given by

$$Sc^{MMR}(i) = \lambda(Sim(S_i, D)) - (1-\lambda)(Sim(S_i, Summ)),$$

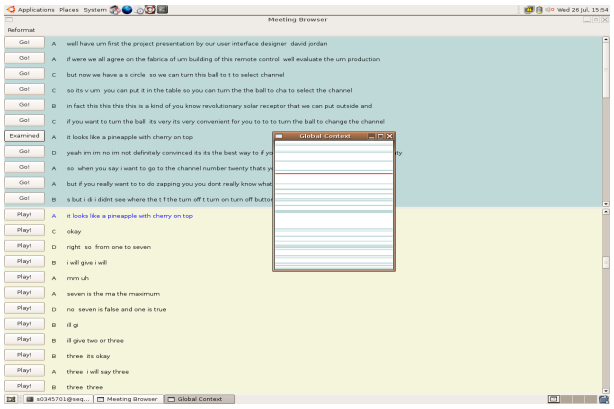

Figure 3: *A meeting browser incorporating extractive summaries*

where D is the average document vector, *Summ* is the average vector from the set of sentences already selected, and λ trades off between relevance and redundancy. *Sim* is the cosine similarity between two documents.

This implementation of MMR uses lambda annealing so that relevance is emphasized while the summary is still short and minimizing redundancy is prioritized more highly as the summary lengthens.

LSA The second text summarization approach implemented was LSA. LSA is a vector-space approach which involves projecting the original term-document matrix to a reduced dimension representation. It is based on the singular value decomposition (SVD) of an $m \times n$ term-document matrix A, whose elements A_{ij} represent the weighted term frequency of term i in document j. In SVD, the term-document matrix is decomposed as follows:

$$A = USV^T$$

where U is an $m \times n$ matrix of left-singular vectors, S is an $n \times n$ diagonal matrix of singular values, and V is the $n \times n$ matrix of right-singular vectors. The rows of V^T may be regarded as defining topics, with the columns representing sentences from the document. Following [Gong and Liu, 2001], summarization proceeds by choosing, for each row in V^T, the sentence with the highest value. This process continues until the desired summary length is reached.

Two drawbacks of this method are that dimensionality is tied to summary length and that good sentence candidates may not be chosen if they do not "win" in any dimension ([Steinberger and Ježek, 2004]). [Steinberger and Ježek, 2004] found one solution, by extracting a single LSA-based sentence score, with variable dimensionality reduction.

We address the same concerns, following the Gong and Liu approach, but rather than extracting the best sentence for each topic, the n best sentences are extracted, with n determined by the corresponding singular values from matrix S. The number of sentences in the summary that will come from the first topic is determined by the percentage that the largest singular value represents out of the sum of all singular values, and so on for each topic. Thus, dimensionality reduction is no longer tied to summary length and more than one sentence per topic can be chosen. Using this method, the level of dimensionality reduction is essentially learned from the data.

Prosodic Features The third summarization system in this experiment uses prosodic features and a simple keyword feature. Feature-based classification approaches have been widely used in text and speech summarization, with positive results ([Kupiec et al., 1995]). In this work we combined textual and prosodic features, using Gaussian mixture models for the extracted and non-extracted classes. The prosodic features were the mean and standard deviation of F0, energy, and duration, all estimated and normalized at the word-level, then averaged over the utterance. The two lexical features were both TFIDF-based: the average and the maximum TFIDF score for the utterance.

For our second feature-based approach, we derived single LSA-based sentence scores ([Steinberger and Ježek, 2004]) to complement the six features described above, to determine whether such an LSA sentence score is beneficial in determining sentence importance. We reduced the original term-document matrix to 300 dimensions; however, Steinberger and Ježek found the greatest success in their work by reducing to a single dimension (Steinberger, personal communication). The LSA sentence score was obtained using:

$$Sc_i^{LSA} = \sqrt{\sum_{k=1}^{n} v(i,k)^2 * \sigma(k)^2},$$

where $v(i,k)$ is the kth element of the ith sentence vector and $\sigma(k)$ is the corresponding singular value.

ROUGE We used the ROUGE evaluation approach ([Lin and Hovy, 2003]), which is based on n-gram co-occurrence between machine summaries and "ideal" human summaries. ROUGE is currently the standard objective evaluation measure for the Document Understanding Conference [12]; ROUGE does not assume that there is

[12] http://duc.nist.gov/

a single "gold standard" summary. Instead it operates by matching the target summary against a set of reference summaries. ROUGE-1 through ROUGE-4 are simple n-gram co-occurrence measures, which check whether each n-gram in the reference summary is contained in the machine summary. ROUGE-L and ROUGE-W are measures of common subsequences shared between two summaries, with ROUGE-W favoring contiguous common subsequences. [Lin and Hovy, 2003] has found that ROUGE-1 and ROUGE-2 correlate well with human judgments. In this experiment, an older version of ROUGE was used which used only n-gram recall. Subsequent experiments used a newer version of ROUGE which calculates recall, precision and f-score, as well as including metrics such as ROUGE-SU4, which allows intervening material to occur in a bigram.

ROUGE Results All of the machine summaries were 10% of the original document length, in terms of the number of dialogue acts contained. Of the four approaches to summarization used herein, the latent semantic analysis method performed the best on every meeting tested for every ROUGE measure with the exception of ROUGE-3 and ROUGE-4. This approach was significantly better than either feature-based approach ($p<0.05$), but was not a significant improvement over MMR. For ROUGE-3 and ROUGE-4, none of the summarization approaches were significantly different from each other, owing to data sparsity. Figure 1 gives the ROUGE-1, ROUGE-2 and ROUGE-L results for each of the summarization approaches, on both manual and ASR transcripts.

The results of the four summarization approaches on ASR output were much the same, with LSA and MMR being comparable to each other, and each of them outperforming the feature-based approaches. On ASR output, LSA again consistently performed the best.

Interestingly, though the LSA approach scored higher when using manual transcripts than when using ASR transcripts, the difference was small and insignificant despite the nearly 30% WER of the ASR. All of the summarization approaches showed minimal deterioration when used on ASR output as compared to manual transcripts, but the LSA approach seemed particularly resilient. One reason for the relatively small impact of ASR output on summarization results is that for each of the 6 meetings, the WER of the summaries was lower than the WER of the meeting as a whole. Similarly, [Valenza et al., 1999] and [Zechner and Waibel, 2000] both observed that the WER of extracted summaries was significantly lower than the overall WER in the case of broadcast news. The table below demonstrates the discrepancy between summary WER and meeting WER for the six meetings used in this research.

Meeting	**Summary WER/%**	**Meeting WER/%**
Bed004	27.0	35.7
Bed009	28.3	39.8
Bed016	39.6	49.8
Bmr005	23.9	36.1
Bmr019	28.0	36.5
Bro018	25.9	35.6
WER Comparison for LSA Summaries and Whole Meetings		

There was no improvement in the second feature-based approach (adding an LSA sentence score) as compared with the first feature-based approach. The sentence score used here relied on a reduction to 300 dimensions, which may not have been ideal for this data.

In general, the comparable performance of LSA and MMR in this research reenforces some of Gong and Liu's key findings. In their work, implementations of LSA and MMR-style summarizers yielded very similar results, prompting the authors to claim that the relatively straightforward interpretation of the MMR algorithm is thus reflected in the more opaque LSA method. In other words, they make the strong claim that the singular vectors of V^T can be interpreted as topics or concepts, and that the LSA summarization method emphasizes relevance and minimizes redundancy.

Further Results In [Murray et al., 2005b], more robust evaluations were carried out to determine whether the ROUGE findings reported above were reliable. Specifically, we elicited human judgments of all of the summaries

STATEMENT	FB1	LSA	MMR	FB2
IMPORTANT POINTS	5.03	4.53	4.67	4.83
NO REDUNDANCY	**4.33**	2.60	3.00	3.77
RELEVANT	4.83	4.07	4.33	4.53
TOPIC SPACE	4.43	3.83	3.87	4.30
REPETITIVE	**3.37**	4.70	4.60	3.83
UNNECESSARY INFO.	**4.70**	6.00	5.83	5.00

Table 23: Human Ratings for 4 Approaches on Manual Transcripts

to see whether or not these judgments would correlate with the ROUGE scores. This subjective evaluation portion of our research utilized 5 judges who had little or no familiarity with the content of the ICSI meetings. Each judge evaluated 10 summaries per meeting, for a total of sixty summaries. In order to familiarize themselves with a given meeting, they were provided with a human abstract of the meeting and the full transcript of the meeting with links to the audio. The human judges were instructed to read the abstract, and to consult the full transcript and audio as needed, with the entire familiarization stage not to exceed 20 minutes.

The judges were presented with 12 questions at the end of each summary, and were instructed that upon beginning the questionnaire they should not reconsult the summary itself. 6 of the questions regarded informativeness and 6 involved readability and coherence, though our current research concentrates on the informativeness evaluations. The evaluations used a Likert scale based on agreement or disagreement with statements, such as the following Informativeness statements:

1. The important points of the meeting are represented in the summary.
2. The summary avoids redundancy.
3. The summary sentences on average seem relevant.
4. The relationship between the importance of each topic and the amount of summary space given to that topic seems appropriate.
5. The summary is repetitive.
6. The summary contains unnecessary information.

Statements such as 2 and 5 above are measuring the same impressions, with the polarity of the statements merely reversed, in order to better gauge the reliability of the answers

Table 1 presents average ratings for the six statements across four summarization approaches on manual transcripts. Interestingly, the first feature-based approach is given the highest marks on each criterion. For statements 2, 5 and 6 the first feature-based approach (FB1) is significantly better than the other approaches. It is particularly surprising that FB1 would score well on statement 2, which concerns redundancy, given that MMR and LSA explicitly aim to reduce redundancy while the feature-based approaches are merely classifying utterances as relevant or not. The second feature-based approach was not significantly worse than the first on this score.

Considering the difficult task of evaluating ten extractive summaries per meeting, we are quite satisfied with the consistency of the human judges. For example, statements that were merely reworded versions of other statements were given consistent ratings. It was also the case that, with the exception of evaluating the sixth statement, judges were able to tell that the manual extracts were superior to the automatic approaches.

Table 2 presents average ratings for the six statements across four summarization approaches on ASR transcripts. The LSA and MMR approaches performed better in terms of having less deterioration of scores when used on ASR output instead of manual transcripts. LSA-ASR was not significantly worse than LSA on any of the 6 ratings. MMR-ASR was significantly worse than MMR on only 3 of the 6. In contrast, FB1-ASR was

STATEMENT	FB1	LSA	MMR	FB2
IMPORTANT POINTS	3.53	**4.13**	3.73	3.50
NO REDUNDANCY	3.40	2.97	2.63	3.57
RELEVANT	3.47	3.57	3.00	3.47
TOPIC SPACE	3.27	3.33	3.00	3.20
REPETITIVE	4.43	4.73	4.70	4.20
UNNECESSARY INFO	5.37	6.00	6.00	5.33

Table 24: Human Ratings for 4 Approaches on ASR Transcripts

significantly worse than FB1 for 5 of the 6 approaches, reinforcing the point that MMR and LSA seem to favor extracting utterances with fewer errors. Figures 2, 3 and 4 depict the how the ASR and manual approaches affect the INFORMATIVENESS-1, INFORMATIVENESS-4 and INFORMATIVENESS-6 ratings, respectively. Note that for Figure 6, a higher score is a worse rating.

In general, ROUGE did not correlate well with the human evaluations for this data. The MMR and LSA approaches were deemed to be significantly better than the feature-based approaches according to ROUGE, while these findings were reversed according to the human evaluations. An area of agreement, however, is that the LSA-ASR and MMR-ASR approaches have a small and insignificant decline in scores compared with the decline of scores for the feature-based approaches. One of the most interesting findings of this research is that MMR and LSA approaches used on ASR tend to select utterances with fewer ASR errors.

3.3.2 Second Experiment: Document Understanding Conference 2005

This section describes a summarization system that was developed and submitted to the Document Understanding Conference (DUC) 2005. Although this system was for text summarization, our participation in DUC 2005 had great ramifications both in terms of our subsequent speech summarization experiments and in terms of deciding on evaluation protocols. A major focus of DUC 2005 was to have a community-wide discussion of summarization evaluation techniques, and so evaluation frameworks such as ROUGE, Basic Elements ([E. Hovy and Fukumoto, 2006]) and Pyramids ([Nenkova and Passonneau, 2004]) were analyzed and discussed.

The Embra (Edinburgh Multi-document Breviloquence Assay) system is based on a Maximal Marginal Relevance (MMR) framework [Carbonell and Goldstein, 1998], where a single extraction score is derived by combining measures of relevance and redundancy of candidate sentences. The system is novel in that it measures relevance and redundancy using a very large latent semantic space. It addresses specificity by detecting the presence or absence of Named Entities in our extract candidates. And it implements a sentence-ordering algorithm to maximize sentence coherence in our final summaries. This attempts to maximise contextual similarity between the original source document and the summary while also grouping sentences based on similarity in the latent semantic space.

A common approach for determining relevance and redundancy in multi-document summarization is to use Maximal Marginal Relevance (MMR), in which candidate sentences are represented as weighted term-frequency vectors which can thus be compared to query vectors to gauge similarity and already-extracted sentence vectors to gauge redundancy, via the cosine of the vector pairs [Carbonell and Goldstein, 1998]. While this has proved successful to a degree, the sentences are represented merely according to weighted term frequency in the document, and so two similar sentences stand a chance of not being considered similar if they don't share the same terms. One way to rectify this is to do Latent Semantic Analysis (LSA) on the matrix first before proceeding to implement MMR, but this still only exploits term co-occurrence *within* the documents at hand.

In contrast, our system attempts to derive more robust representations of sentences by building a large semantic space using LSA on a very large corpus. While researchers have used such large semantic spaces to aid in automatically judging the coherence of documents ([Foltz et al., 1998, Barzilay and Lapata, 2005]), to our knowledge this is a novel technique in summarization. Using a concatenation of Aquaint and DUC 2005 data (100+

```
for each sentence in document:
        for each word in sentence:
                get word vector from semantic model
        average word vectors to form sentence vector
        sim1 = cossim(sentence vector, query vector)
        sim2 = highest(cossim(sentence vector, all extracted vectors))
        score = λ*sim1 - (1-λ)*sim2
        if sentence contains multiple named entities:
                if granularity == 'specific':
                        weight score higher
                else if granularity == 'general':
                        weight score lower
        else:
                do not weight score
extract sentence with highest score
repeat until desired length
```

Table 25: Sentence extraction algorithm

million words), we utilized the Infomap tool[13] to build a semantic model based on latent semantic analysis (LSA) of the corpora. LSA ([Landauer and Dumais, 1997]) utilizes singular value decomposition of a term/document matrix, with the documents here being newspaper articles. The decomposition and projection of the matrix to a lower-dimensionality space (in this case, 100 dimensions) results in a semantic model based on underlying term relations. There are numerous ways to query the model, such as finding the most closely related words to a given word or deriving a word vector for a given word. Using such word vectors, a given sentence can be represented as a vector which is the average of its constituent word vectors. This sentence representation can subsequently be fed into an MMR-style algorithm. Our implementation of the algorithm (see Table 25) uses λ annealing following [Murray et al., 2005a]. λ decreases as the summary length increases, thereby emphasizing relevance at the outset but increasingly prioritizing redundancy removal as the process continues.

DUC 2005 Results DUC 2005 set a single query-oriented, multi-document summarisation task for newswire data. There were 50 topic clusters to be summarised with respect to a short topic query consisting of a 1 to 4 sentence description of an information need. An additional constraint indicated whether the summary should be specific or general. There were 31 participating systems. For the results reported here, individual system scores are averaged over topic clusters.

All DUC systems were evaluated manually for responsiveness and five measures of linguistic proficiency. Human evaluation scores for responsiveness (Rsp, defined below), grammaticality (LQ1), non-redundancy (LQ2), referential clarity (LQ3), focus (LQ4), and structure/coherence (LQ5) can be found in Table 26. The Embra system performance is better than mean and median system scores for the responsiveness measure and for three of the five linguistic quality measures (grammaticality, non-redundancy and focus). It is just below mean and median scores for structure/coherence. In terms of referential clarity, the system rank falls to 28 out of 31.

Responsiveness is defined as *the amount of information in the summary that helps to satisfy the information need expressed in the topic*. The fact that the system does fairly well on this measure suggests that the latent semantic model does a good job of accounting for relevance and redundancy.

Evaluation Issues from DUC 2005 The ROUGE results from DUC 2005 displayed a *flatness of scores*, i.e. the metric was not able to discern well between various systems. Its correlation with human judgments of relevance was low at microaveraged and macroaveraged levels, with the latter being somewhat higher. According to ROUGE, the vast majority of summarization systems were not significantly different from one another. ROUGE-2 and ROUGE-SU4 were the best performing metrics, but it was found that ROUGE-SU4, which allows intervening material in the bigram, did not perform any better than simple bigram comparison. Similarly, Basic Elements ([E. Hovy and Fukumoto, 2006]), which uses comparison of minimal semantic units rather than n-gram overlap, did not add any benefit to just using ROUGE-2.

[13] `http://infomap.stanford.edu/`

	Rsp	LQ1	LQ2	LQ3	LQ4	LQ5
BLine	1.98	4.26	4.68	4.58	4.50	4.00
Min	1.38	2.60	3.96	2.16	2.38	1.60
Mean	2.40	3.76	4.40	2.94	3.11	2.12
StDev	0.30	0.43	0.21	0.43	0.35	0.35
Median	2.44	3.86	4.44	2.98	3.16	2.10
Embra	***2.44***	***3.92***	***4.48***	***2.38***	***3.24***	***2.00***
Max	2.78	4.34	4.74	4.14	3.94	3.24
UpBnd	4.67	4.81	4.91	4.93	4.89	4.76

Table 26: Embra scores compared to average system performance for human metrics.

Pyramids ([Nenkova and Passonneau, 2004]) is a summarization evaluation method that requires manual annotation of "semantic content units" (SCUs), which are typically phrase-level units. The details of the method can be read in more detail in [Nenkova and Passonneau, 2004], and the results of using Pyramids at DUC 2005 are briefly reported here. First, a fundamental drawback of the Pyramids method is that it requires a large amount of manual annotation. It is far from being an automated method like ROUGE or Basic Elements, and it is only really feasible to use it when there are a large number of annotators available as with DUC. Second, annotators participating in the 2005 DUC Pyramid evaluation found the task very difficult, as 40% of submitted annotations required moderate or major corrections by Columbia University. Finally, Pyramids suffered from the same flatness of scores problem as ROUGE did; it was unable to differentiate the performance of systems very well.

While Embra was not submitted to DUC 2006, we did follow the continuing community discussion of evaluation protocols. In 2006, Pyramid results were better and this was partly attributed to annotators developing more familiarity with the framework. Also in 2006, it was found that Basic Elements performed worse than ROUGE. It has been hypothesized that this is due to more systems implementing sentence compression techniques which often result in ungrammatical text, which subsequently causes Basic Elements to be unable to derive structure from the text.

3.3.3 Third Experiment: Speech Features and LSA Centroid Approaches

In the LSA approach described in the first experiment and in [Murray et al., 2005a], the singular value decomposition (SVD) is performed on a matrix of *tf.idf* values. The intuition behind this experiment is that additional features beyond *tf.idf* can be put into a matrix which is to undergo SVD. Specifically, we are interested in prosodic features such as dialogue act length, structural features such as position within a meeting, and speaker related features such as identifying areas of high speaker activity and determining speaker status within the meeting. A second summarization system uses only lexical features and uses the sentence representation method from the Embra system. The two systems are described below.

Unlike the first experiments, which extracted 10% of the meeting's dialogue acts to form a summary, this experiment uses a much shorter summary length of 350 words. This is a more challenging task because it requires greater precision, and it also allows us to more easily elicit human judgments in the future if the summaries are not so long. This experiment was again carried out on the ICSI corpus.

Speech Features with SVD In previous summarization work on the ICSI corpus [Murray et al., 2005a, Murray et al., 2005b], Murray et al. explored multiple ways of applying latent semantic analysis (LSA) to a term/document matrix of weighted term frequencies from a given meeting, a development of the method in [Gong and Liu, 2001]. A central insight to the present work is that additional features beyond simple term frequencies can be included in the matrix before singular value decomposition (SVD) is carried out. We can use SVD to project this matrix of features to a lower dimensionality space, subsequently applying the same methods as used in [Murray et al., 2005a] for extracting sentences.

The features used in these experiments included features of speaker activity, discourse cues, listener feedback, simple keyword spotting, meeting location and dialogue act length (in words).

For each dialogue act, there are features indicating which speaker spoke the dialogue act and whether the same speaker spoke the preceding and succeeding dialogue acts. Another set of features indicates how many speakers are active on either side of a given dialogue act: specifically, how many speakers were active in the preceding and succeeding five dialogue acts. To further gauge speaker activity, we located areas of high speaker interaction and indicated whether or not a given dialogue act immediately preceded this region of activity, with the motivation being that informative utterances are often provocative in eliciting responses and interaction. Additionally, we included a feature indicating which speakers most often uttered dialogue acts that preceded high levels of speaker interaction, as one way of gauging speaker status in the meeting. Another feature relating to speaker activity gives each dialogue act a score according to how active the speaker is in the meeting as a whole, based on the intuition that the most active speakers will tend to utter the most important dialogue acts.

The features for discourse cues, listener feedback, and keyword spotting were deliberately superficial, all based simply on detecting informative words. The feature for discourse cues indicates the presence or absence of words such as *decide*, *discuss*, *conclude*, *agree*, and fragments such as *we should* indicating a planned course of action. Listener feedback was based on the presence or absence of positive feedback cues following a given dialogue act; these include responses such as *right*, *exactly* and *yeah*. Keyword spotting was based on frequent words minus stopwords, indicating the presence or absence of any of the top twenty non-stopword frequent words. The discourse cues of interest were derived from a manual corpus analysis rather than being automatically detected.

A structural feature scored dialogue acts according to their position in the meeting, with dialogue acts from the middle to later portion of the meeting scoring higher and dialogue acts at the beginning and very end scoring lower. This is a feature that is well-matched to the relatively unstructured ICSI meetings, as many meetings would be expected to have informative proposals and agendas at the beginning and perhaps summary statements and conclusions at the end.

Finally, we include a dialogue act length feature motivated by the fact that informative utterances will tend to be longer than others.

The extraction method follows [Steinberger and Ježek, 2004] by ranking sentences using an LSA sentence score. The matrix of features is decomposed as follows:

$$A = USV^T$$

where U is an $m \times n$ matrix of left-singular vectors, S is an $n \times n$ diagonal matrix of singular values, and V is the $n \times n$ matrix of right-singular vectors. Using sub-matrices S and V^T, the LSA sentence scores are obtained using:

$$Sc_i^{LSA} = \sqrt{\sum_{k=1}^{n} v(i,k)^2 * \sigma(k)^2},$$

where $v(i,k)$ is the kth element of the ith sentence vector and $\sigma(k)$ is the corresponding singular value.

Experiments on a development set of 55 ICSI meetings showed that reduction to between 5–15 dimensions was optimal. These development experiments also showed that weighting some features slightly higher than others resulted in much improved results; specifically, the discourse cues and listener feedback cues were weighted slightly higher.

LSA Centroid The second summarization method is a textual approach incorporating LSA into a centroid-based system ([Radev et al., 2001]). The centroid is a pseudo-document representing the important aspects of the document as a whole; in the work of [Radev et al., 2001], this pseudo-document consists of keywords and their modified *tf.idf* scores. In the present research, we take a different approach to constructing the centroid and to representing sentences in the document. First, *tf.idf* scores are calculated for all words in the meeting. Using these scores, we find the top twenty keywords and choose these as the basis for our centroid. We then perform LSA

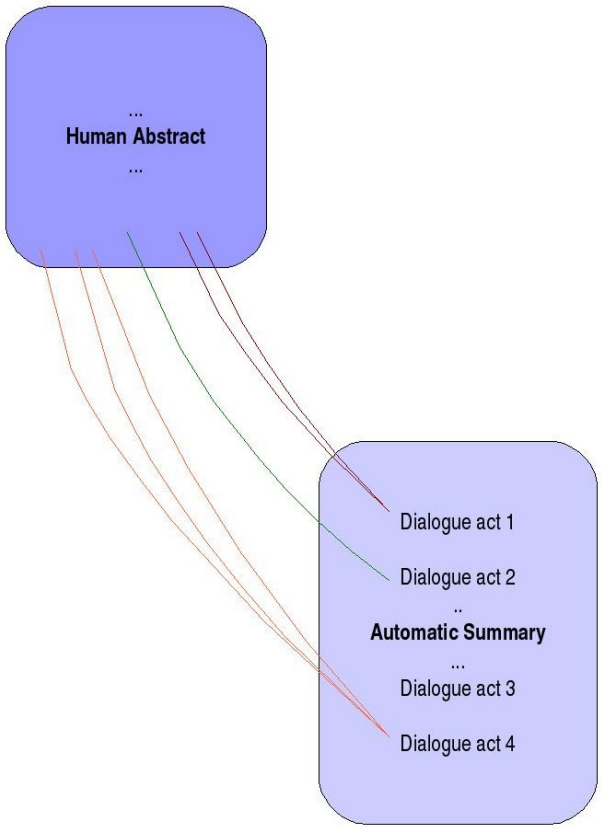

Figure 4: *Evaluation with Weighted Precision*

on a very large corpus of Broadcast News and ICSI data, using the Infomap tool[14]. Infomap provides a query language with which we can retrieve word vectors for our twenty keywords, and the centroid is thus represented as the average of its constituent keyword vectors ([Foltz et al., 1998, Hachey et al., 2005]).

Dialogue acts from the meetings are represented in much the same fashion. For each dialogue act, the vectors of its constituent words are retrieved, and the dialogue act as a whole is the average of its word vectors. Extraction then proceeds by finding the dialogue act with the highest cosine similarity with the centroid, adding this to the summary, then continuing until the desired summary length is reached.

Combined The third summarization method is simply a combination of the first two. Each system produces a ranking and a master ranking is derived from these two rankings. The hypothesis is that the strength of one system will differ from the other and that the two will complement each other and produce a good overall ranking. The first system would be expected to locate areas of high activity, decision-making, and planning, while the second would locate information-rich utterances. This exemplifies one of the challenges of summarizing meeting recordings: namely, that utterances can be important in much different ways. A comprehensive system that relies on more than one idea of importance is ideal.

Evaluation I - Weighted Precision For the ICSI corpus, we have manual abstractive summaries and manual extractive summaries. We also have links between these two types of summaries, so that transcript dialogue acts can

[14]http://infomap.stanford.edu

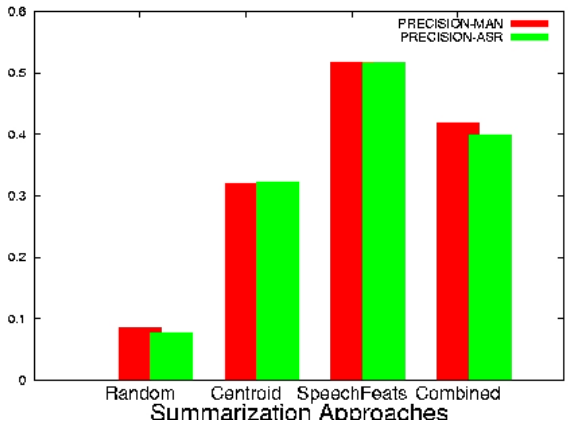

Figure 5: *Weighted Precision Results on Test Set*

be linked to an abstract sentence if the dialogue acts supports the sentence. An abstract sentence can be linked with more than once dialogue act, and likewise a dialogue act can support more than one abstract sentence. This many-to-many mapping of dialogue acts to abstract sentences allows us to evaluate our extractive summaries according to how often each annotator linked a given extracted dialogue act to a summary sentence. This is somewhat analogous to Pyramid weighting ([Nenkova and Passonneau, 2004]), but with dialogue acts as the SCUs. In fact, we can calculate weighted precision, recall and f-score using these annotations, but because the summaries created are so short, we focus on weighted precision as our central metric. For each dialogue act that the summarizer extracts, we count the number of times that each annotator links that dialogue act to a summary sentence. For a given dialogue act, it may be that one annotator links it 0 times, one annotator links it 1 time, and the third annotator links it two times, resulting in an average score of 1 for that dialogue act. The scores for all of the summary dialogue acts can be calculated and averaged to create an overall summary score. Figure 4 illustrates how weighted precision is calculated.

Evaluation II - ROUGE ROUGE scores, based on n-gram overlap between human abstracts and automatic extracts, were also calculated for comparison ([Lin and Hovy, 2003]). ROUGE-2, based on bigram overlap, is considered the most stable as far as correlating with human judgments, and this was therefore our ROUGE metric of interest. ROUGE-SU4, which evaluates bigrams with intervening material between the two elements of the bigram, has recently been shown in the context of the Document Understanding Conference (DUC)[15] to bring no significant additional information as compared with ROUGE-2. Results from [Murray et al., 2005b] and from DUC 2005 also show that ROUGE does not always correlate well with human judgments. It is therefore included in this research in the hope of further determining how reliable the ROUGE metric is for our domain of meeting summarization.

Results For weighted precision, the speech features approach was easily the best and scored significantly better than the centroid and random approaches (ANOVA,$p<0.05$), attaining an averaged weighted precision of 0.52. The combined approach did not improve upon the speech features approach but was not significantly worse either. The randomly created summaries scored much lower than all three systems.

The superior performance of the speech features approach compared to the LSA centroid method closely mirrors results on the ICSI development set, where the centroid method scored 0.23 and the speech features approach scored 0.42. For the speech features approach on the test set, the best feature by far was dialogue act length. Removing this feature resulted in the precision score being nearly halved. This mirrors results from

[15] http://duc.nist.gov

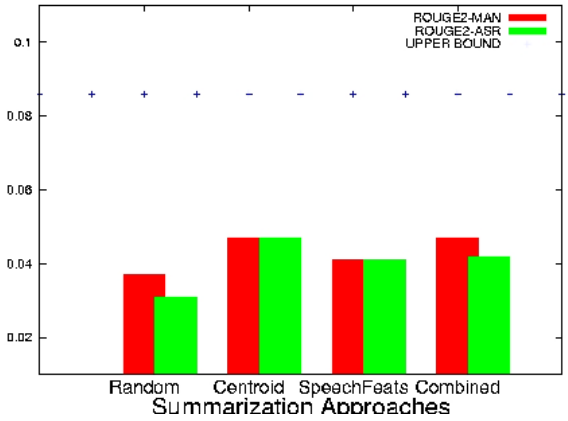

Figure 6: *ROUGE-2 Results on Test Set*

[Maskey and Hirschberg, 2005], who found that the length of a sentence in seconds and its length in words were the two best features for predicting summary sentences. Both the simple keyword spotting and the discourse cue detection features caused a lesser decline in precision when removed, while other features of speaker activity had a negligible impact on the test results.

Interestingly, the weighted precision scores on ASR were not significantly worse for any of the summarization approaches. In fact, the centroid approach scored very slightly higher on ASR output than on manual transcripts. In [Valenza et al., 1999] and [Murray et al., 2005a] it was similarly found that summarizing with ASR output did not cause great deterioration in the quality of the summaries. It is not especially surprising that the speech features approach performed similarly on both manual and ASR transcripts, as many of its features based on speaker exchanges and speaker activity would be unaffected by ASR errors. The speech features approach is still significantly better than the random and centroid summaries, and is not significantly better than the combined approach on ASR.

The ROUGE results greatly differed from the weighted precision results in several ways. First, the centroid method was considered to be the best, with a ROUGE-2 score of 0.047 compared with 0.041 for the speech features approach. Second, there were not as great of differences between the four systems according to ROUGE as there were according to weighted precision. In fact, the random summaries of manual transcripts are not significantly worse than the other approaches, according to ROUGE-2. Neither the combined approach nor the speech features approach is significantly worse than the centroid system, with the combined approach generally scoring on par with the centroid scores.

The third difference relates to summarization on ASR output. ROUGE-2 has the random system and the combined system showing sharp declines when applied to ASR transcripts. The speech features and centroid approaches do not show declines. Random summaries are significantly worse than both the centroid summaries ($p<0.1$) and speech features summaries ($p<0.05$). Though the combined approach declines on ASR output, it is not significantly worse than the other systems.

To get an idea of a ROUGE-2 upper bound, for each meeting in the test set we left one human abstract out and compared it with the remaining abstracts. The result was an average ROUGE-2 score of .086.

ROUGE-1 and ROUGE-SU4 show no significant differences between the centroid and speech features approaches.

There is no significant correlation between macroaveraged ROUGE and weighted precision scores across the meeting set, on both ASR and manual transcripts. The Pearson correlation is 0.562 with a significance of $p < 0.147$. The Spearman correlation is 0.282 with a significance of $p < 0.498$. The correlation of scores across each test meeting is worse yet, with a Pearson correlation of 0.185 ($p<0.208$) and a Spearman correlation of 0.181 ($p<0.271$).

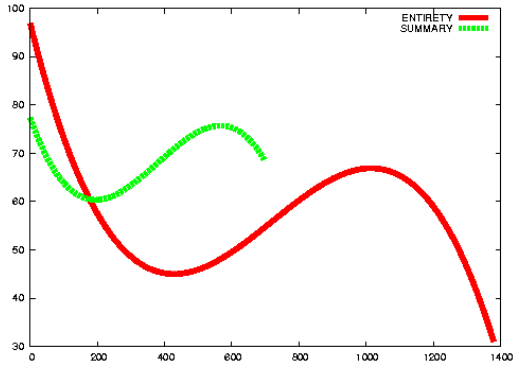

Figure 7: *Sample Dialogue Act and Summary Contours (first prosodic method)*

3.3.4 Fourth Experiment: Dialogue Act Compression

As reported above, one of the best features for detecting informative dialogue acts is the length of the dialogue act, either in number of seconds or number of words. This creates a problem in that we are trying to create very short summaries (350 words) and yet our units of extraction are quite long. Subsequently, a given summary may contain only 10 or so dialogue acts. This experiment tries to address this problem by implementing various methods of automatically compressing dialogue acts.

Unfortunately, the fragmented and disfluent nature of meeting speech means that importing text compression techniques is not feasible. Meeting dialogue acts cannot be reliably parsed, and we are thus limited as to how we can both determine the essential components of a given dialogue act and have a resulting compression that is readable. This section therefore explores the use of prosody in compressing informative dialogue acts from meeting speech. More specifically, the techniques described below compress the dialogue acts by trying to preserve the original pitch contour as much as possible in the compressed dialogue act. The simple intuition behind this method is that *prosody is meaning* ([Steedman, 2000]) and that preserving this aspect of the prosody may preserve a great deal of the meaning as well.

Two methods of using prosody for speech compression are described below. They are first evaluated subjectively by humans grading on both informativeness and readability criteria, alongside human-authored gold-standards and random baseline compressions. The second evaluation is edit distance, objectively measuring the string distance between the automatic approaches and the gold-standards. In addition to the prosodic and random approaches, a simple text compression method was implemented and included for this edit distance evaluation. The compression rate is between 0.65 and 0.70 for all of the automatic compression methods.

Prosodic Compression Methods The first prosody method begins by breaking the utterance into prosodic phrases or chunks. The primary cue for phrase boundary is pause length, with pauses of 100 ms or more being considered a boundary. A secondary method is to look for instances of pitch reset which would signal the beginning of a new prosodic phrase. More specifically, we are looking for areas where the pitch falls to a low level for at least 300 ms before rising sharply again, with the fall-rise pattern signalling the pitch declination of one phrase and the beginning of another. We first attempt to locate the boundaries using only pause, as it is considered more reliable, but if we are unable to break the dialogue act into at least 3 chunks, we revert to looking at pitch reset as well. Once the prosodic phrases are located, the overall pitch slope for each phrase is measured. We then begin an iterative process, wherein for each phrase we measure the pitch slopes of its constituent words and select the word whose slope is closest to that of the phrasal slope. If a phrase has no more than two words, we skip it altogether as it is likely to be a disfluent fragment. We continue the iterative process until the desired number of words has been selected for the compression.

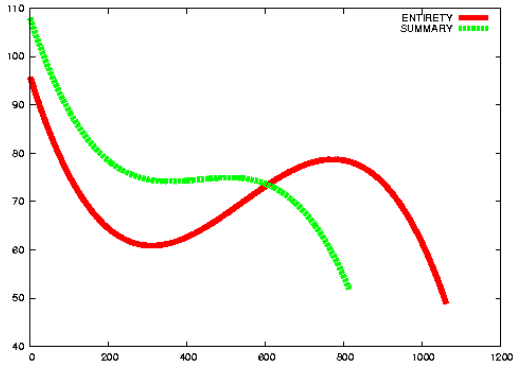

Figure 8: *Sample Dialogue Act and Summary Contours (second prosodic method)*

Figure 7 shows cubic regressions for the pitch contours of the following utterance and summary pair:

Original: *So given these um these features or or these these examples um critical examples which they call support f- support vectors then um gien a new example if the new example falls um away from the boundary in one direction then it's classified as being a part of this particular class*

Compression: *So given these features or these examples critical examples which they call support vectors then given a new example if new example falls boundary in one direction then being a part of this particular class*

The second prosodic method is more crude and does not depend on recognizing phrase boundaries. Instead, the pitch contour for the entire dialogue act is represented as a vector of F0 values. Compression proceeds by deleting words one at a time, based on how large an effect each word's deletion has on the pitch contour. For each iteration of the procedure, each word has its F0 values deleted from the pitch vector and replaced with interpolated values between its former neighbouring words. This new pitch vector is then compared with the original pitch vector by using cosine similarity. The word with the highest cosine similarity is deleted, as the removal of its F0 values had little effect on the overall pitch contour. Again, the procedure continues until the desired length is reached.

Essentially, the two prosodic methods are working from opposite directions, one iteratively selecting words while the other is iteratively eliminating words. There are significant procedural differences, however, as the latter method does not use phrasal information and thus would not ignore short fragments as the former method would. This second method also relies on overall pitch vector similarity, which may not be as reliable as measuring slope at the phrasal and word levels.

Figure 8 shows cubic regressions for the pitch contours of the following utterance and summary pair:

Original: *And the interesting thing is that even though yes it's a digits task and that's a relatively small number of words and there's a bunch of digits that you train on it's just not as good as having a a l- very large amount of data and training up a a a nice good big HMM*

Compression: *And interesting thing is that though yes it's digits task and that's relatively small words and there's bunch digits you train on it's just not good as having a large amount and training up a nice good big HMM*

Comparison Methods For the second evaluation scheme described below, we implemented a simple text compression method for comparison. As in the methods described above, we began by deleting filled pauses and repetitions. We then assigned each word in the dialogue act a *tf.idf* score, a metric which gives high ranks to words that are frequent within a document but rare across multiple documents. We select the words with the highest *tf.idf* scores until the desired compression length is reached. This text compression method is quite simple but nevertheless would give a reasonable expectation of high informativeness.

To assess baseline performance, we randomly select the desired number of words and present them in the original order.

The gold standard for compression is human-authored compressions. Manual compressions were made with

a compression rate between 60% and 70%. The manual compressions were restricted to using only words from the original dialogue act and had to be presented in the original order, as with the automatic methods. The slightly wider window for the compression rate is because it is not feasible to require human annotators to compress an utterance to a precise percentage of the original.

Subjective Evaluation Human judges were presented with the output of four compression methods on the test set, for a total of 120 compressions to be evaluated. These four methods were random baseline compressions, human-authored gold-standard compressions, and the two prosodic compression methods. The judges were asked to rate each compression for two criteria, informativeness and readability. The ratings were made on a 1-5 Likert scale with 1 being 'Very Poor' and 5 being 'Very Good.'

When rating a given compression in terms of its informativeness, judges were asked to keep in mind whether the compression retained the most important parts of the original utterance and refrained from including irrelevant or unnecessary parts of the original. They were instructed that this is a distinct and separate rating from readability, so that a compression may score high on informativeness and still do very poorly on readability.

When rating a given compression in terms of its readability, judges were asked to consider whether the compression seemed grammatical and fluent relative to the original and whether the compression was generally readable. The term *relative* was included in the instructions because a compression which is an ungrammatical fragment should not be scored very low if the original utterance was also an ungrammatical fragment, for example.

Edit Distance The second method of evaluation is edit distance, which utilizes our human-authored compressions as a gold-standard for an objective comparison. The edit distance between two strings is defined as $1-(I+D+S)/R$, where R is the number of words in the reference string and I, D and S are insertions, deletions and substitutions, respectively. This metric thus objectively measures how close an automatically compressed string comes to the ideally compressed string. For this evaluation, four compression approaches were measured against the reference string, with the four approaches being random, text-based, and two prosodic approaches.

Results The inter-annotator agreement was very good for subjective informativeness ratings, with the correlation of macroaveraged scores above 0.9 for each annotator pair. The manual compressions were rated significantly higher than the others ($p<0.05$), with an average informativeness score of 4.65. Both of the prosodic approaches were significantly better than random ($p<0.05$) but were not significantly different from one another. The first prosodic approach had an average informativeness score of 3.69 and the second prosodic approach had an average of 3.82. The random compressions averaged 2.08 in terms of informativeness.

The inter-annotator agreement was again very good for the readability judgements, with correlations above 0.9 for each annotator pair. The significant effects are the same as those of the informativeness scores, with the manual compressions rating significantly higher than the other approaches ($p<0.05$) and the prosodic approaches being signficantly better than random ($p<0.05$) but not significantly different from one another. The manual compressions had an average readability score of 4.6, the first prosodic approach averaged 2.93, the second prosodic approach averaged 3.15, and the random compressions averaged 1.77 in terms of readability. Interestingly, while the random and prosodic approaches had readability scores significantly lower than their informativeness scores, the manual compressions scored comparably on both readability and informativeness.

The most striking aspect of the edit distance results is that the *tf.idf* method performed only at the level of the random method. The prosodic approaches were significantly better ($p<0.05$), with an average edit distance of 0.56 and 0.53, respectively. The *tf.idf* and random approaches each had an average edit distance of 0.44.

Though the second prosodic method was thought to be cruder than the first, it performed slightly but not significantly better in terms of both readability and informativeness. Future work may combine the two methods in order to optimize the compression results.

3.4 Abstractive Summaries

Automated abstractive summarization utilizes a fundamentally different approach to generate summaries. Following [Sparck-Jones, 1998], the summarization process is globally divided into three parts:

- *I*: source *interpretation* to source representation
- *T*: source representation *transformation* to summary represenatation
- *G*: summary text *generation* from summary representation

For the representation of both the source meeting(s) and the summary, we have developed a domain ontology based on the W3C semantic web standard OWL[16] and the WordNet lexical database[17], containing over 60,000 ontological categories. For the realization of the above steps, automatic components will have to perform the following steps:

1. Analysis of the meeting recordings, integrating various results from this and other WPs, such as automatic speech recognition (ASR), addressee and dialog act recognition, gesture and meeting act recognition. On a technical level, the analysis process results in a structure of ontological instances, drawn from the categories of the domain ontology, and relational properties connecting these instances.
2. Relevance detection and restructuring of the ontological representation. In this step, the ontology graph representing the meeting is transformed by identifying subgraphs that represent irrelevant information and merging suitable subparts to a yield a compact summary graph.
3. The use of natural language generation (NLG) techniques to realize the internal representation as coherent text.

This division defines a straight-forward processing pipeline that allows us to realize system components independently. This fact virtually averts the natural deferment our work is experiencing through the heavy dependency on results from other work packages. We are able to implement a prototype relying on hand annotated data while it is possible to successively replace hand annotation with automatic components.

Intrinsic evaluation methods like ROUGE compare the terms that appear in the source and the summary, but it is obvious that such a technique is only applicable for summarization methods that extract their summary contents directly from the original sources. In the case of abstractive summaries, where a NLG component is used to produce the summary text from an internal representation, it can not be expected that the terms appearing in the summary are the same as in the source. For instance, parts of the summary might get realized with synonyms of the words originally used by the meeting participants. In such a case, although the same information is conveyed, term-level comparisons spuriously lead to poor results.

Therefore, we have opted for an extrinsic evaluation method. (It is worth noting that we plan to use this method for the extractive summaries as well, allowing us to compare the value of both approaches.) For the extrinsic evaluation method, we draw upon work on the so-called *browser evaluation test* (BET) [Wellner et al., 2005] conducted in Work Package 6.

The BET provides a framework for the comparison of arbitraty setups of a meeting browser. It consists of a series of experiments involving test persons to gather objective data about how well a user can find information in a recorded meeting. In each experiment, the test person is presented a version of a meeting browser together with a list of pairs of statements about the meeting. These statements result from a hand-collection of so-called *observations of interest* from the meeting where one of the two statements is true and the other one is false. The experimentee uses the meeting browser to find out for each pair which is the true statement. A time limit of half of the actual meeting duration asserts that the experimentee can not just play back an entire meeting in its full length

[16] http://www.w3.org/TR/owl-ref
[17] http://wordnet.princeton.edu

and work on the observations of interest afterwards. The final result of the test is based on the number of correctly identified observations of interest in the given period of time.

We will apply the BET to compare one browser with a summary component to one without such a component and also to compare browsers with summary components on the basis of different summarization techniques. This will give us insight in the usefullness of meeting summaries in general, as well as a lineup of the practical quality of the different summarization approaches.

A setup without any summarization component in the browser is well-suited to define a baseline value that we should be able to outperform. A topline value can be created by using a browser setup containing a summary component that displays human-authored summaries.

3.5 Evaluation Issues Overview

Throughout the experiments described above, ROUGE did not seem to correlate well with human judgments. Specifically, ROUGE seems unable to discern between the varying performances of different summarization systems. ROUGE may be of limited use for development, when a new component is added to a summarization system and we want a very rough idea of whether any improvement has been gained. That is, it may be suitable for tracking performance *within* a system, but is far from ideal for discerning between the performances of multiple systems.

For development purposes, we instead propose to use the weighted precision method, which is analogous to Pyramids. Because we have these multiple summary links for both the ICSI and AMI corpus, it is easy to compute the weighted precision scores when a new batch of summaries are created. This has thus been adopted as the development evaluation metric for AMI extractive summarization work.

However, we ultimately want evaluations of how well our summaries aid a user in navigating meeting content via a meeting browser interface. In other words, an *extrinsic* evaluation which, rather than directly measuring the information content of a given summary, judges the usefulness of the summary in helping to perform a task. Thus we will evaluate summaries in the context of a larger Browser Evaluation Test (BET). Various types of functionality for navigating a meeting will be available to a user and we will determine which tools are most useful in answering questions related to observations of interest from the meeting.

We are also developing an extrinsic evaluation for the AMI scenario meetings which involves re-running the fourth and final meeting of a series, with new participants. This involves the new participants preparing for their meeting by individually browsing the content of the first three meetings via a meeting browser. They must build off of the planning and decision-making that the previous participants made in the first three meetings, and so being able to efficiently navigate these prior meetings is critical. We hypothesize that extractive summaries will allow the new participants to quickly discover the important information and decision results from those meetings, helping them to more quickly and efficiently complete their overall design and marketing task.

3.6 Results and Discussion

We have implemented numerous summarization systems, and so far have generated a large amount of evidence to support the hypothesis that using speech-specific features for speech summarization is advantageous. Particularly, features such as dialogue act length and speaker status are useful in locating informative dialogue acts from a meeting. Up to this point we have concentrated on the ICSI meeting corpus data for extractive summarization and are transititioning to the AMI data, where we will implement our systems on the new data and determine whether there are large and significant performance results on differing types of corpora.

Because informative dialogue acts tend to be very long and our summaries are relatively brief, we have also implemented dialogue act compression methods and compared them. It has been shown that a prosodic approach to compression performs very well, signficantly better than a simple *tf.idf* word ranking. Thus, we have shown that prosody is not only useful for extraction but also post-extraction cleanup.

As described in above and throughout this section overall, we have also decided on an extraction summarization evaluation paradigm. The extrinsic evaluation will involve recruiting new meeting partipants and re-running the

fourth meeting of a series, with the users able to navigate the content of previous meetings using a meeting browser containing extractive summaries.

3.7 Outlook

Current extractive summarization work has involved applying the findings of our ICSI experiments to new summarization approaches on the AMI corpus. One promising system under development is unsupervised and relies largely on speaker status and dialogue act duration. We first determine who the leader of a meeting is by combining several scores: who speaks the most dialogue acts in a meeting, who speaks the largest number of seconds in the meeting, and who speaks the largest number of words. Once we have determined the meeting leader, we create a preliminary summary (currently 350 words in length) of dialogue acts from that speaker only. We locate the meeting leader's most informative dialogue acts by ranking dialogue acts on two criteria: length in seconds and length in number of words. Once we have created the preliminary summary, we extract informative dialogue acts of the other speakers by combining two criteria: length in words and similarity to the preliminary summary content. The latter type of similarity is determined by representing both the preliminary summary dialogue acts and the candidate dialogue acts using latent semantic analysis, in the style of the Embra system. We can subsequently use cosine measurements to gauge similarity between candidate sentences and the summary so far. This approach is currently being evaluated intrinsically.

For abstractive summarization, the most important task is the implementation of automatic components to replace hand annotated data. We have realized a large-scale domain ontology, particularily tailored to the requirements of the AMI domain, that builds the backbone for all computational processing. In addition, we are about to finish a first version of the language generation component which will give us the means at hand for extrinsic evaluation. We will have the possibility to continually re-evaluate the quality of our system by running a BET experiment whenever a new automated component is inserted into the pipeline, replacing formerly used hand data. The details of the BET evaluation are currently begin planned among involved partners.

4 Chunking

4.1 Introduction

4.1.1 The Task

Chunking is the task of "dividing text into syntactically related non-overlapping groups of words" [Tjong Kim Sang and Buchholz, 2000, p.127]. It can be considered a form of partial parsing, and was partly motivated by the psychological evidence that the human parser works on non-recursive cores of major phrases [Abney, 1991]. Abney both pioneered the idea of parsing by chunks and built a knowledge-intensive, rule-based chunker [Abney, 1996].

More recently, chunking has been reformulated as a task similar to part-of-speech (POS) tagging [Ramshaw and Marcus, 1995]. In this approach, if only noun chunks are required, then the words from a corpus might be marked up using three tags: B for the initial word of a noun chunk, I for a non-initial word of a noun chunk, and O for a word outside any chunk. Tagging more chunk types requires a larger tag set that has B and I categories for each chunk type; for instance, a noun and verb chunker might use a representation that has the tag set {B-N, I-N, B-V, I-V, O}. This reformulation allows the many learning approaches that have been developed for sequential classification and which are familiar from POS tagging to be applied directly to the problem of chunking [Osborne, 2000, Osborne, 2002].

The first community competition for chunking technology was held in CONLL-2000. In the chunking shared task, which used part of WSJ articles as the base material, the organisers created training and test data for not just noun and verb chunks but a wide variety of chunk types by converting some of the syntactic annotation distributed as part of the PTB into tags using the basic reformulation described above.[18] The agreed evaluation metrics for the task were precision, recall, and an F score ($F_{\beta=1}$, which gives precision and recall equal weight). The best performance among the eleven participating systems was F1=93.48 [Kudo and Matsumoto, 2000]. The main choices that developers of chunkers have exercised are the set of features to use when classifying, exactly how to reformulate the chunks as a tag set, and which classifier to use. Common features used include the current word being tagged; words in windows of various sizes extending into the left context of the current word, and possibly also into the right context; POS tags for all of these words including the current word; and chunk tags for words in the left (or right, if chunking backward) context windows. Alternative reformulations typically augment the basic dichotomy between chunk-initial (B) and non-initial (I) words with distinctions for the I category that differentiate based on position in the chunk, particularly picking out chunk-final words.

After the CONLL 2000 shared task, various efforts have been spent on improving chunking performance on the common data, e.g., F1 reached 93.91 with SVM voting [Kudo and Matsumoto, 2001], 93.57 with generalised Winnow [Zhang et al., 2002], 93.71 with filtering-ranking perceptron learning [Carreras et al., 2005], 94.39 with additional unlabelled data [Ando and Zhang, 2005], etc.

All chunking work to date has focused solely on performance improvements for WSJ texts. However, one paper, [Osborne, 2002], has considered the effects of training on data drawn from the SWBD corpus, which contains conversational dialogues, but testing on WSJ data. Osborne carried out this test not because he was interested in the effects of training on one corpus and testing on another, but as a way of estimating the effects of noisy training data. That is, he was using the known degraded grammaticality associated with speech as an approximation for noise in training data. For instance, using section 2 of the SWBD corpus for his training data, a maximum entropy-based classifier, and a feature set that included the last four letters of the current word plus POS labels for the current and two following words, he achieved a performance of F1=82.97. Although he did originally intend his data for these purposes, his results can be reconstructed as more data points about the effects of training on one corpus for testing another from a fairly dissimilar genre. They are not directly comparable to our own results that train on SWBDdata and test on WSJ because we have used a larger training set but done less to maximise performance. However, when reinterpreted in this way, his results are broadly compatible with our own.

[18] The script used for this conversion is available from `http://ilk.uvt.nl/~sabine/chunklink/chunklink_2-2-2000_for_conll.pl`. valid on 2006-07-14.

4.1.2 The Data

We use four different corpora in this work, first three of which are drawn from PTB [Marcus et al., 1993], the last from AMI corpus [Carletta et al., 2006], labelled in our figures and tables as follows:

- PTB/WSJ (CONLL) This is the same data that was used in CONLL-2000: that is, sections 15-18 of the WSJ portion of the PTB as training data, and section 20 as test data.
- PTB/BROWN This is taken from the portion of the Brown corpus that has been labelled with syntactic annotation as part of the PTB. It consists of a collection of textual sources ranging in genre, but largely drawn from different types of fiction. For our test data, we chose one text from each of the annotated subsets representing different text genres.
- PTB/SWBD This is taken from the portion of the Switchboard corpus that has again been labelled with syntactic annotation as part of the PTB. We used sections 2 and 3 as training data and section 4 as test data. The switchboard corpus consists of telephone conversations between pairs of unfamiliar participants in which the participants were given an overall topic of conversation.
- AMI This is a portion of the section of the AMI meeting corpus that consists of four-party group discussions in which the participants simulate a workplace design team. Two short meetings (IS1008a and ES2008a) are used as test data, and two long meetings (IS1008b and ES2008b) as training data.

Because the first three of these corpora come from the PTB, we simply derive chunks for them using the scripts distributed for CONLL-2000. This leaves the AMI meeting corpus, for which no PTB portion is available. To address this issue, we hand-annotated training and test data sets from the AMI corpus for most of the chunk types used in CONLL-2000, omitting PRT, LST, and UCP because they are extremely rare. [19]

From these corpora, we used WSJ (CONLL), SWBD, and AMI for training and all of the corpora for testing. The training sets vary considerably in size, as can be seen from table 27. Note that two of the four corpora are from textual sources and two consist of transcribed speech. All four are arguably from different genres, although one would generally expect the text corpora to share more characteristics with each other than with the speech corpora, and the speech corpora to share more characteristics with each other than with the text corpora. This means, for instance, that whether one's target corpus is a text corpus or a speech corpus, it would be reasonable to expect better performance when training on another corpus of the same basic type than the opposite basic type.

Table 27: Data details (in words, 1K = 1,000)

data	training	test	size
WSJ (CONLL)	220K	49K	medium
SWBD	1,054K	223K	large
BROWN	–	20K	
AMI	13.6K	6.6K	small

4.2 Method

When building a chunker, the first step is to choose a classifier. There were many possible classifiers we could have chosen, and if we were interested in maximising performance alone, we might wish to try them all. The three below represent a reasonable spread of choices. Each comes from a different theory and is both publicly available and well-implemented.

[19] The annotation was performed using a version of the named entity annotation tool reconfigured for chunks, from the NITE XML Toolkit, which is available at `http://www.ltg.ed.ac.uk/NITE/` or `http://sourceforge.net/projects/nite/`.

MAXENT/MXPOST [20] This is a classical statistical POS taggers based on maximum entropy theory [Ratnaparkhi, 1996]. Although it was written in Java about ten years ago, it is fast and demands less computing resource than the other two.

SVM/YAMCHA-TinySVM [21] This classifier is based on support vector machines (SVM) [Vapnik, 1998]. An SVM-based chunker won the first place in the CONLL 2000 task [Kudo and Matsumoto, 2000]. This particular implementation produced even better result than that [Kudo and Matsumoto, 2001].

CRF/CRF++ [22] This is one of the post-CONLL models that achieve better performance CRF [Sha and Pereira, 2003], based on Conditional Random Fields theory [Lafferty et al., 2001]. Though the experiments were carried out on NP chunking, the performance is comparable with other chunkers on the same task.

For the features, we refrain from using some complex ones: for words, we use unigram of current word and two words before and after current word; for chunk tags, we use unigram of last two tags.[23] We did not use POS information so as to make things less complicated because for any new data there will be neither directly available tagger, nor manual annotation. The reason we choose such simple features is, on the one hand, to ensure fair comparison across classifiers,[24] and on the other, to make computation as easy as possible, keeping in mind limited resources available for online processing in some real-world applications.

4.3 Evaluation

The evaluation is done with the same Perl script as the CONLL shared task, which is available at `http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt`. The performance is reported in F_1 measure, which is a harmonic mean of precision and recall.

4.4 Results

4.4.1 Chunkers Trained on Three Corpora with Three Classifiers

We trained a couple of chunkers with the above three classifiers and simple feature configuration on the training sets of three corpora. And all the chunkers were tested on the four test sets (with additional selection of BROWN as test data). The results are given in Table 28.

First we can see from the table that despite its relatively modest computational demands, the maximum entropy classifier performs best most of the time when the test and training data come from different sources. When the test and training data come from the same source, the result of the maximum entropy classifier is very close to the best results from the CRF one.

It is also very clear that the chunkers work best when trained on the data from the same source. This is not beyond expectation. But there is some difference between them: F1 is around 88 for WSJ, 90 for SWBD, and 81 for AMI. This could be easily explained by the difference in the training data size. From Table 27, compared with the medium-sized WSJ (CONLL), SWBD is much larger and AMI is much smaller. Further experiments on the effect of training data size on chunking performance will be presented below (§ 4.4.2).

But it is not that easy to compare the performance for the chunkers on unmatched data (i.e., trained on the data from one source, tested on the data from others). Can we explain by genre difference in terms of the distinction between written texts and spoken dialogue? For the chunkers trained on written WSJ, the performance on BROWN

[20] Available at `ftp://ftp.cis.upenn.edu/pub/adwait/jmx/jmx.tar.gz`.

[21] Available at `http://chasen.org/~taku/software/yamcha/` and `http://chasen.org/~taku/software/TinySVM/`.

[22] Available at `http://chasen.org/~taku/software/CRF++/`.

[23] It was not untill very late that we found that features for MXPOST target tags are unigram of last tag and bigram of last two tags. But the result from this is only slightly better than that using unigrams of last two tags. So the results and discussions below should still hold though the comparisons are not strictly fair.

[24] This is the default feature manipulation for MXPOST. It is possible to extend through concatenation, like [Osborne, 2000].

Table 28: Chunking performance (F1) for three classifiers trained on three training corpora, tested on the four test sets

	MAXENT	SVM	CRF
Training on WSJ/CoNLL			
WSJ	88.35	87.85	88.55
BROWN	83.29	80.37	81.57
SWBD	71.06	70.62	71.49
AMI	62.84	58.34	61.91
Training on SWBD			
WSJ	76.70	75.39	77.78
BROWN	82.43	79.55	81.20
SWBD	89.93	90.96	91.82
AMI	75.08	71.90	73.50
Training on AMI			
WSJ	61.41	51.54	57.29
BROWN	62.62	49.62	55.16
SWBD	70.09	66.88	69.28
AMI	81.72	78.96	82.02

is much closer than that on SWBD and AMI. For the chunkers trained on AMI data, the performance on SWBD is much closer than that on WSJ and BROWN. The spoken-written distinction can partly be supported by the chunk distributions in the data, as shown in figure 9.

But for the chunkers trained on SWBD, the performance on BROWN is much closer than that on WSJ and AMI, which could not be simply explained by the spoken-written distinction. We further calculated Kullback-Leibler divergence (or relative entropy) – a measure of distance between two distributions p and q – for the chunk distributions. It is defined as $D(p||q) = \sum_{x \in X} p(x) \log p(x)/q(x)$. For our case, $p, q \in$ [WSJ, SWBD, BROWN, AMI], x is the chunk type. The result is given in Table 29.

Table 29: Kullback-Leibler divergence for chunk distributions

q ↓ p →	WSJ (CONLL)	BROWN	SWBD	AMI
WSJ (CONLL)	0.00	0.02	0.61	0.98
BROWN	0.02	0.00	0.27	0.54
SWBD	0.18	0.12	0.00	0.15
AMI	0.27	0.21	0.06	0.00

Since Kullback-Leibler divergence indicates the inefficiency of assuming that the distribution is q when the true distribution is p [Cover and Thomas, 1991], and is asymmetric with regard to p and q (as can been seen from both the definition and table 29), if we look at the table horizontally, we can see that it would be more efficient

- to approximate BROWN with WSJ than SWBD and AMI;
- to approximate WSJ with BROWN than SWBD and AMI; and
- to approximate SWBD with AMI than WSJ and BROWN.

And although it seems more efficient to approximate BROWN with SWBD than WSJ and AMI, it is similarly efficient (or inefficient) to approximate any of WSJ, BROWN and AMI with SWBD. This explains why for the chunkers trained on SWBD the performance on BROWN is much closer than that on WSJ and AMI, instead of that the performance

on AMI should be closer than that on WSJ and BROWN. If we look at the table vertically, we can see that it would be more efficient

- to approximate WSJ with BROWN than with SWBD or AMI,
- to approximate BROWN with WSJ than with SWBD or AMI,
- to approximate SWBD with AMI than with WSJ or BROWN, and
- to approximate AMI with SWBD than with WSJ or BROWN.

This should help on how to choose from relavant data when we have to build chunkers without matched data.

4.4.2 No Data Like More Data?

So far we have investigated the effects of employing different training corpora on chunking performance. There is another open and similarly important question: when annotating new data from the target corpus, how much data is required to reach a given level of performance? Having the answer to this question, together with the cost of annotation, would aid those planning work that relies on chunking technology.

To answer the question, we run an experiment for WSJ and SWBD data that starts with 10K words of training data and add 10K words at each iteration, testing performance at each data round. The experiment uses the same data set as before for SWBD, which we again label as SWBD, but instead of restricting ourselves to the 220K words of WSJ training data that were employed in CONLL, here we used the full 1,173K words available from the PTB. We chose our maximum entropy classifier for this investigation because it provided the best all-round performance for the lowest computational costs. But this time we ran two feature set versions: our original, plus one that concatenates POS to the word features used. The POS tags do not come from some automatic tagger, but from the PTB annotation. This is to avoid unnecessary uncertainty. But in many cases we will have to use the output of some POS tagger. The results are shown in Figure 10.

In the figure, the learning curves are all quite similar. POS tags help for WSJ data when the training data set is small, but these gains nearly disappear as the training data set size increases. However, for the SWBD corpus, the gains remain almost constant throughout the curve. The figure shows that after 20-25 iterations, or 200-250K words, the potential performance gains of annotating more data are really quite low.

Although the amount of training data that we used was in this range or higher for the WSJ and SWBD corpora, our AMI training data set was much smaller than this, as it consisted of 13.6K words. Figure 11 shows the result of a similar experiment intended to blow up the initial section of the learning curve by having the iterations start at 1K words and add 1K words per round. This time we exclude the POS tagged conditions, but include a curve for the AMI data. The figure shows that the learning curve for the AMI data is very similar to those for the other corpora, at least in this initial section, and gives no reason to believe that the relationship between performance and data requirements is any different for this corpus.

4.4.3 General Discussions

We have presented an empirical study on cross-corpora syntactic chunking, training on three different corpora and testing on four. We also plot the learning curves in order to estimate how much data would be enough for training a chunker. Our work is a promising beginning for the deployment of chunking in a wider range of speech and language technologies.

Our experimentation shows

- that if the very highest performance is required, there is no substitute for training chunkers on plentiful annotation from the target corpus.

  In our experiments on data from the AMI meeting corpus, the best performance we attained using training data from a different corpus could be easily beaten by using annotation from the target corpus for an

extremely modest amount of data (ca 4.3K words).[25] This also echoes a similar statement from statistical parsing [Gildea, 2001, p.200]: "a small amount of matched training data appears to be more useful than a large amount of unmatched data." But annotation is expensive. Performance gains diminish greatly once 200-250 K words of annotation are available for the target corpus, suggesting there is little point in investment beyond this.

- that if there is no annotation from the same source available, but only some annotated data from other sources (maybe in other form, like syntactic trees in the PTB), then reuse a more similar data would be more beneficial. The performance levels that can be attained by training on existing data from other corpora may well suffice for many end applications, particularly those in speech where imperfect speech recognition means that safeguards against incorrect interpretation must be built in anyway.
- that chunking spontaneous speech does not raise any particular difficulties or require any change of chunking strategies previously derived for newspaper texts. Actually it is the distributional difference that matters more to chunking than the distinction between written text and spoken dialogue where there is more noise from disfluencies. Hence chunking is very much noise-tolerant. This further confirms previous work on the effect of noisy materials on chunking [Osborne, 2002].

4.5 Outlook

There are still some open problems. In order to determine the best strategies for developing a chunker on new types of data, we must develop some measure (like K-L divergence) for the distance between source and target corpora that predicts how well a chunker trained on the source will perform on the target. In future work we should also consider the possibility of training chunkers using data drawn from multiple sources and determine how to predict the amount of corpus-specific annotation required to raise performance to whatever level an application requires.We need work on some general framework for domain/genre adaption, like [Daumé III and Marcu, 2006]. Another more challenging task would be to find out how semi-supervised techniques could help chunking through learning from additional unlabelled data, like [Ando and Zhang, 2005].

Acknowledgement

Special thanks go to Adwait Ratnaparkhi and Taku Kudo for their providing the wonderful softwares publicly available.

[25] From Table 28, the best performance for AMI data while training on non-AMI data is F1=75.08 (training on SWBD with MXPOST). This performance could be achieved by the chunker trained on AMI data at the amount of 4.3K words or so, as can be seen from Figure 11.

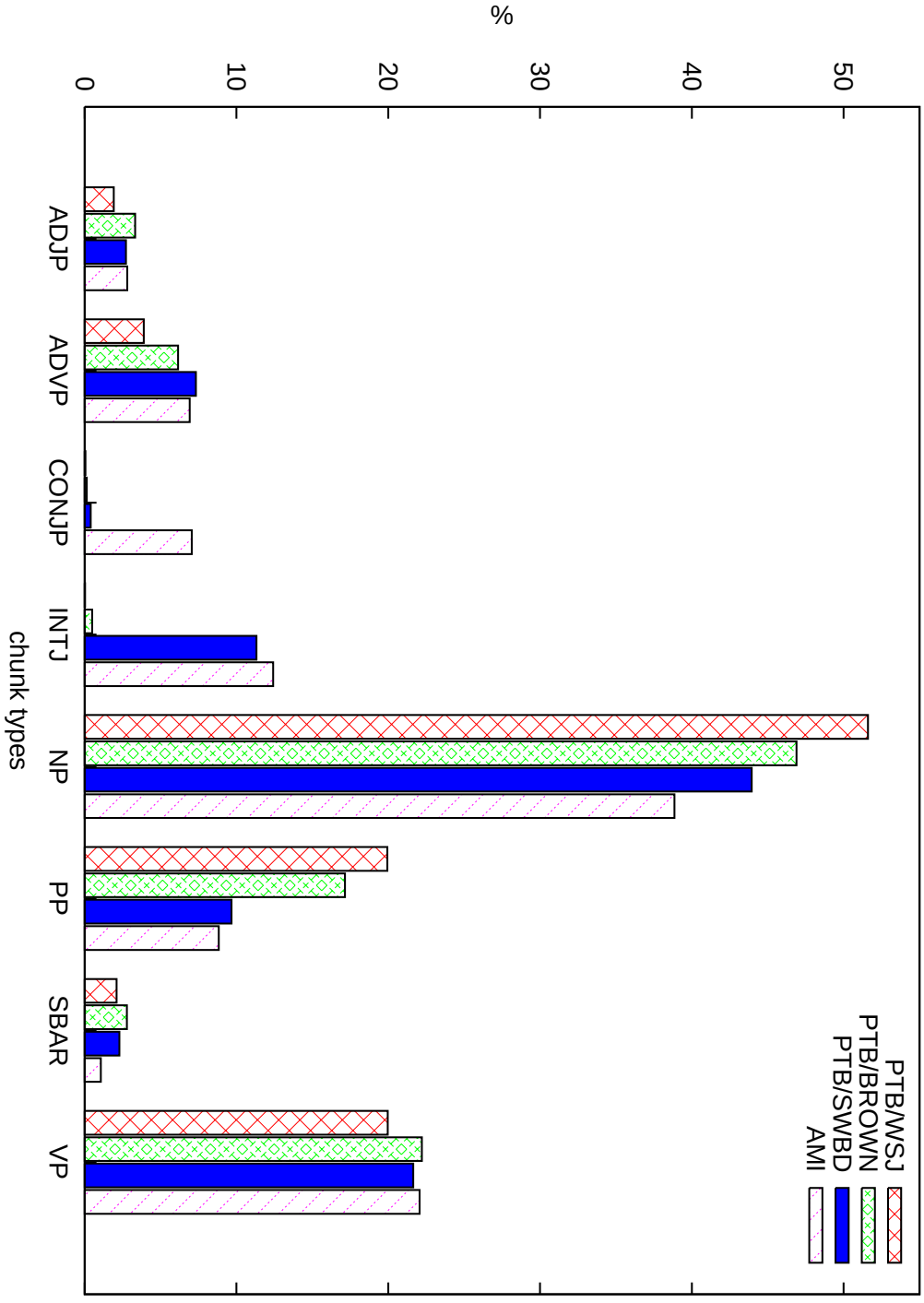

Figure 9: chunk distribution in the data

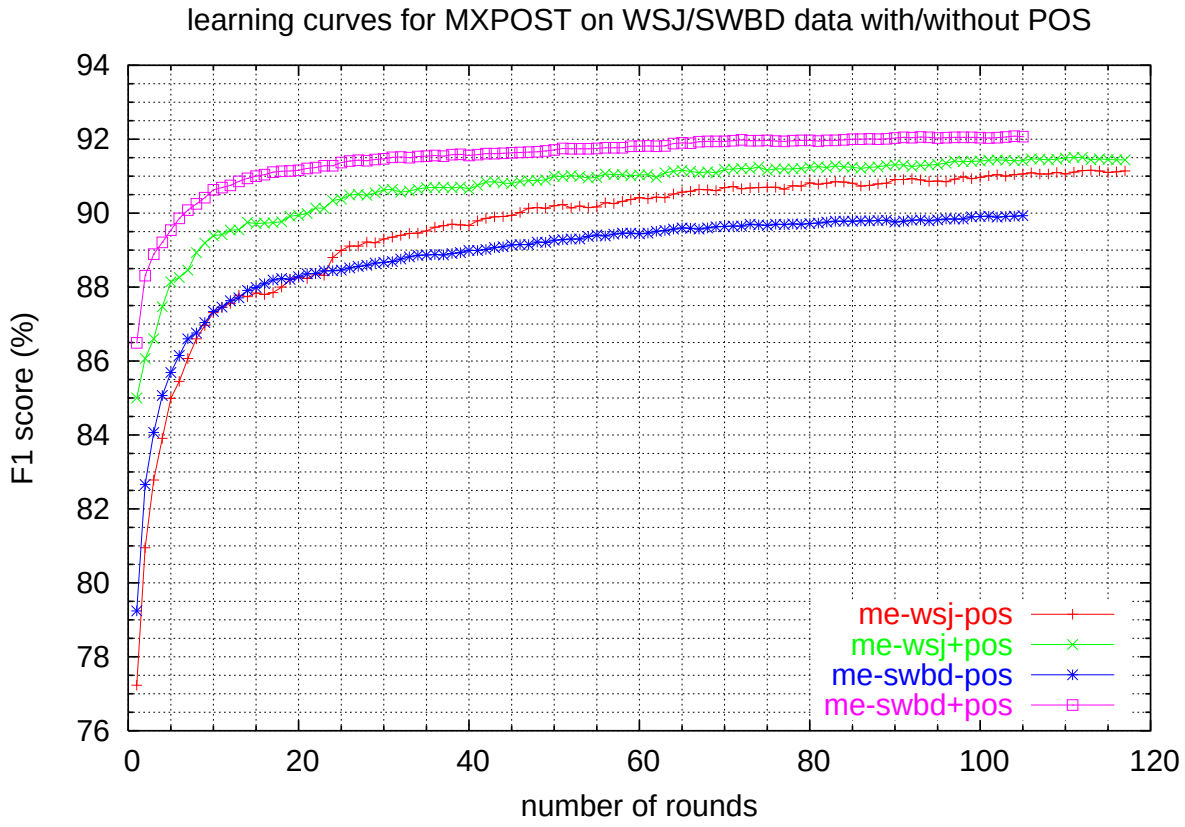

Figure 10: MXPOST on WSJ (full) and SWBD, 10K words per round

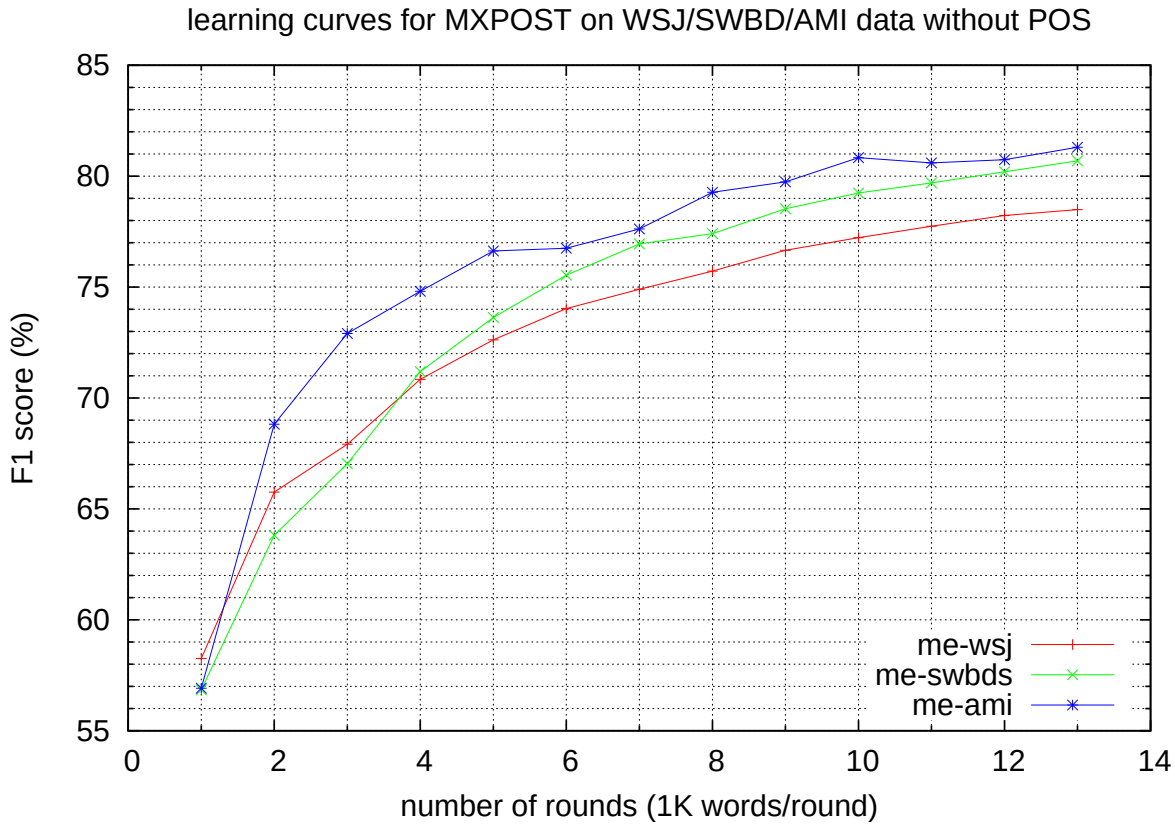

Figure 11: MXPOST on WSJ, SWBD and AMI, 1K words per round

5 Named Entity Identification

5.1 Introduction

5.1.1 Task Definition

The named entity (NE) task involves identification of words or word sequences that may be classified as proper names, or as certain other categories such as monetary expressions, times and dates. This is not a straightforward problem. While 'Wednesday 1 September' is clearly a date, and 'Alan Turing' is a personal name, other strings, such as 'the day after tomorrow', 'South Yorkshire Beekeepers Association' and 'Nobel Prize' are more ambiguous. The task is defined by 'AMI named entity guidelines'[26], which should be read as an addendum to the NIST 1999 NE recognition task definition[27], version 1.4. According to the definition the following NE tags would be correct:

<DATE>Wednesday 1 September</DATE>
<PERSON>Alan Turing</PERSON>
the day after tomorrow
<ORGANIZATION>South Yorkshire Beekeepers Association</ORGANIZATION>
Nobel Prize

'The day after tomorrow' is not tagged as a date, since only *absolute* time or date expressions are recognised; 'Nobel' is not tagged as a personal name, since it is part of a larger construct that refers to the prize. Similarly, 'South Yorkshire' is not tagged as a location since it is part of a larger construct tagged as an organisation.

A set of named entities by the AMI guideline is essentially a superset of the NIST definition, and includes artefacts relevant to meetings such as furniture, drawing, and recording devices. Figure 12 shows the NE hierarchy, where tag sets for the AMI guidelines and for the NIST definition are highlighted. The former consists of 10 types of named entities, while the latter includes more than 20 types.

5.1.2 Named Entity Annotation

Following the AMI guidelines, named entities were manually annotated on 117 hand transcripts of meetings. Their statistics are provided in Table 30. They were split into 100 meetings for development of the automatic NE identification system and 17 meetings for evaluation of the system. A typical meeting transcript contained roughly five thousand words and 150 named entities. They represented three to four percents of the entire words, characterising their sparseness in meetings[28].

5.2 Method

We tried two approaches on this task. First, a finite state model based statistical named entity identification system was built from the development data. It explicitly modelled bigram constraints for transitions between NE types and words, compensating for the fundamental sparseness of bigram tokens on a vocabulary set. The further technical detail may be found in [Gotoh and Renals, 2000a].

The second follows the approach taken by CONLL 2003 shared task [Tjong Kim Sang and De Meulder, 2003]. Similar to chunking, the task is reformulated as sequential tagging. Therefore any sequential classifier could be employed. In fact we used MXPOST [Ratnaparkhi, 1996] again. Evaluation was done with the common script `http://www.cnts.ua.ac.be/conll2003/ner/bin/conlleval` and reported with precision, recall and F1 score.

[26] `http://wiki.idiap.ch/ami/NamedEntities`
[27] `http://www.nist.gov/speech/tests/ie-er/er_99/doc/ne99_taskdef_v1_4.ps`
[28] In comparison, 10 types of named entities account for roughly 9% of broadcast news transcripts [Gotoh and Renals, 2000a].

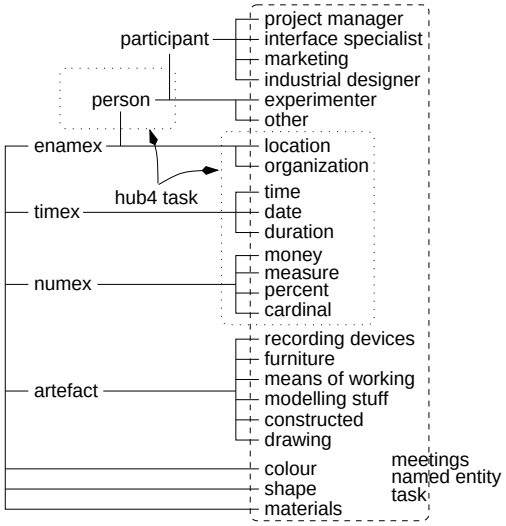

Figure 12: This figure shows the hierarchy of named entities. A tag set for 1999 NIST definition is given by dotted lines, while a tag set for the AMI guidelines is shown by the dashed line.

	development	evaluation
#meetings	100	17
total #words	475791	88651
total #NEs	13411	2930
total #NE words	16966	3602

Table 30: Statistics for named entity annotated meeting data.

5.3 Evaluation

NE identification systems are evaluated using an unseen set of evaluation data: the hypothesised NEs are compared with those annotated in human-generated reference transcripts. In this situation there are two possible types of error: *type*, where an item is tagged as the wrong kind of entity, and *extent*, where the wrong number of word tokens are tagged. For example,

<LOCATION>South Yorkshire</LOCATION> Beekeepers Association

has errors of both kinds, *type* and *extent*, since the ground truth for this excerpt is

<ORGANIZATION>South Yorkshire Beekeepers Association</ORGANIZATION> .

These two error types each contribute 0.5 to the overall error count, and precision (P) and recall (R) can be calculated in the usual way. A weighted harmonic mean ($P\&R$), sometimes called the F-measure [van Rijsbergen, 1979], is often calculated as a single summary statistic

$$P\&R = \frac{2RP}{R+P} .$$

The $P\&R$ score implicitly deweights missing and spurious identification errors compared with incorrect identification errors [Makhoul et al., 1999]. To alleviate this problem, an alternative measure, referred to as the slot error rate (SER), can be calculated as

$$SER = \frac{I+M+S}{C+I+M}$$

where C, I, M, and S denote the numbers of correct, incorrect, missing, and spurious identifications. Using this notation, precision and recall scores may be obtained by

$$R = \frac{C}{C+I+M}$$
$$P = \frac{C}{C+I+S}$$

Evaluation of spoken NE identification is more complicated than for text, since there will be speech recognition errors as well as NE identification errors (i.e., the reference tags will not apply to the same word sequence as the hypothesised tags). This requires a word level alignment of the two word sequences, which may be achieved using a phonetic alignment algorithm developed for the evaluation of speech recognisers. Once an alignment is obtained, the evaluation procedure outlined above may be employed, with the addition of a third error type, *content*, caused by speech recognition errors. The same statistics (P and R) can still be used, with the three error types contributing equally to the error count.

	R	*P*	*P&R*	*SER*
type	0.566	0.688	0.621	0.575
extent	0.650	0.791	0.714	0.491

Table 31: Baseline results (Sheffield) for NE *type* and *extent*.

5.4 Results

Table 31 shows the baseline results by the Sheffield system, whose brakedown for individual NEs is also summarised in Table 32. It achieved a *P&R* score of 66.8%, by weighting equally on NE *type* and *extent*. A lower score (62.1%) for the NE *type* was compensated by a higher score (71.4%) for the NE *extent*. A major failure came from confusion between number expressions (i.e., <DURATION>, <MEASURE>, <PERCENT>, and <CARDINAL>). Further, the system was unable to identify many <CONSTRUCTED> tags and <DRAWING> tags.

The same system achieved 89.2% *P&R* for 1998 Hub-4E named entity evaluation of news broadcasts [Gotoh and Renals, 2000a]. The decline of performance is not very surprising because

- broadcast news system was build from an NE annotated corpus of roughly one million words, while the meeting system was build from a dataset with less than a half million words;
- 9% of transcribed words in broadcast news were named entities; this number went down to below 4% for meetings;
- broadcast news involved annotation of 10 types of named entities, while the task for meeting data was concerned with 25 types;

All of the above contributed the sparseness of named entities in meeting data, which was counted as a significant drawback for the most of statistical approaches. On the other hand, they have their own advantage over grammar based systems. It required months to code the statistical system for broadcast news. It took for a week to port the system into meeting data domain, provided with carefully annotated development data.

Table 33 gives the preliminary results from MXPOST, without any tuning or adaption.

5.5 Outlook

Possible next steps might be: to follow the general data partition; to improve performance; to deal with non-senario data; etc.

	type				*extent*			
	C	*I*	*M*	*S*	*C*	*I*	*M*	*S*
PROJECT MANAGER	6	2	7	0	7	1	7	0
INTERFACE SPECIALIST	9	6	26	0	15	0	26	0
MARKETING	24	11	11	0	33	2	11	0
INDUSTRIAL DESIGNER	22	3	6	0	24	1	6	0
EXPERIMENTER	2	0	0	0	2	0	0	0
OTHER	109	35	21	29	141	3	21	29
LOCATION	11	0	10	0	11	0	10	0
ORGANIZATION	12	0	8	0	12	0	8	0
TIME	12	6	0	0	15	3	0	0
DATE	7	3	4	0	10	0	4	0
DURATION	83	15	11	111	90	8	11	111
MONEY	54	30	10	0	72	12	10	0
MEASURE	117	136	42	0	232	21	42	0
PERCENT	59	2	1	30	60	1	1	30
CARDINAL	410	46	143	61	436	20	143	61
RECORDING DEVICES	8	1	4	29	9	0	4	29
FURNITURE	0	0	2	0	0	0	2	0
MEANS OF WORKING	160	3	62	101	158	5	62	101
MODELLING STUFF	6	0	2	0	6	0	2	0
INCIDENTAL	0	0	0	0	0	0	0	0
CONSTRUCTED	7	12	141	0	19	0	141	0
DRAWING	174	17	324	33	190	1	324	33
COLOUR	131	5	42	19	125	11	42	19
SHAPE	83	3	40	0	84	2	40	0
MATERIALS	152	2	17	0	154	0	17	0
total	1658	338	934	413	1905	91	934	413

Table 32: Baseline results (Sheffield) by individual named entities. *C*, *I*, *M*, and *S* denote the numbers of correct, incorrect, missing, and spurious identifications.

	Precision	Recall	$F_{\beta=1}$
CARDINAL	59.07%	69.62%	63.91
COLOUR	85.31%	68.54%	76.01
CONSTRUCTED	25.00%	1.25%	2.38
DATE	41.67%	35.71%	38.46
DRAWING	69.71%	23.69%	35.36
DURATION	94.37%	61.47%	74.44
EXPERIMENTER	0.00%	0.00%	0.00
FURNITURE	0.00%	0.00%	0.00
INCIDENTAL	0.00%	0.00%	0.00
INDUSTRIAL_DESIGNER	0.00%	0.00%	0.00
INTERFACE_SPECIALIST	0.00%	0.00%	0.00
LOCATION	90.91%	47.62%	62.50
MARKETING	28.95%	23.91%	26.19
MATERIALS	92.90%	91.81%	92.35
MEANS_OF_WORKING	0.00%	0.00%	0.00
MEASURE	80.39%	27.80%	41.31
MODELLING_STUFF	0.00%	0.00%	0.00
MONEY	70.97%	46.81%	56.41
ORGANIZATION	68.75%	55.00%	61.11
OTHER	84.14%	73.94%	78.71
PERCENT	86.15%	90.32%	88.19
PROJECT_MANAGER	0.00%	0.00%	0.00
RECORDING_DEVICES	0.00%	0.00%	0.00
SHAPE	85.90%	53.17%	65.69
TIME	77.78%	38.89%	51.85
Overall	71.46%	44.44%	54.80

Table 33: NER results with MXPOST

6 Topic Segmentation

6.1 Topic Segmentation and Labeling in AMI Corpus

6.1.1 Introduction

This study concerns how to segment a lengthy conversational record into a number of smaller segments and how to label each locally coherent segment automatically. It aims to provide the right level of details for users to interpret the meaning of a conversation beyond the individual utterance level. Our interest to the problem is two-fold: First, topic segmentation and labeling provides the right level of details for users to interpret what has transpired and locate relevant information in a multiparty dialogue. For example, upper management can efficiently identify the discussion of target user groups of a product by browsing the topic hierarchies as in Fig 13. Second, developing automatic topic segmentation and labeling components for multiparty dialogue lends support to computer supported collaborative work applications, where group meeting records are automatically processed in order to extract information for summarization, question answering and providing thumbnail views on mobile devices.

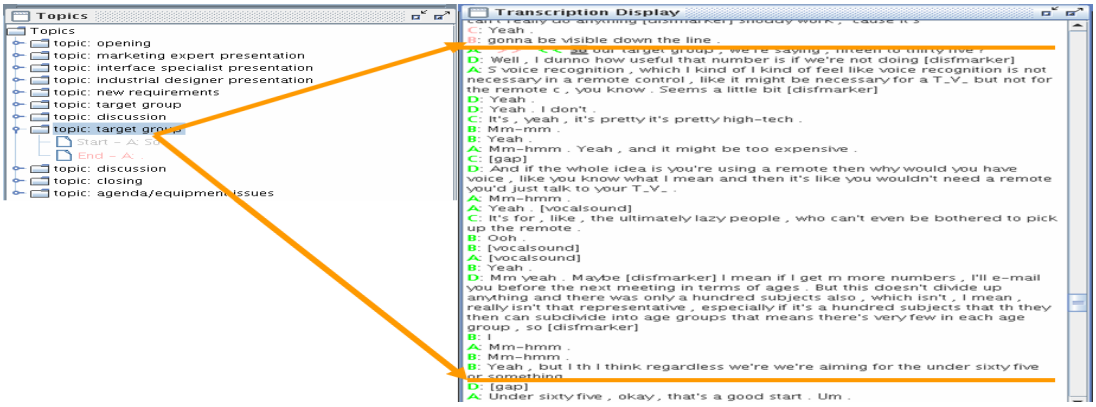

Figure 13: *Example of topic segmentation in a produce design meeting.*

Past research has developed segmentation models to divide a long sequence of utterances into smaller segments based on on two notions of local coherence: information coherence and intentional coherence. Information coherence, on the one hand, is determined by consecutive utterances that are semantically cohesive [Halliday and Hasan, 1976]. Intentional coherence, on the other hand, is determined by utterances that maintain unchanged user intentions [Grice, 1969]. In fact, [Morris and Hirst, 1991] has provided empirical evidences on how the intentional coherence and information coherence are isomorphic in hand annotations. [29] Therefore, in this study, we focus on identifying features that can characterize topic boundaries in the human annotation data.

Past research has explored the effect of a variety of features on characterizing topic boundaries. For example, [Hearst, 1997] has studied lexical cohesion and proposed the TextTiling algorithm, an unsupervised approach that hypothesizes boundaries as points where the lexical cohesion score changes significantly. [Stokes et al., 2004] and [Galley et al., 2003] have also examined the use of lexical cohesion in hypothesizing segment boundaries in broadcast news transcripts and spontaneous speech. Recent advance in statistical text classification has inspired research to cast the segmentation task as a binary classification task. Various combinations of features have been proposed to train the classification models, e.g., prosodic cues [Grosz and Hirschberg, 1992, Litman and Passoneau, 1995], lexical features (n-grams) and discourse cues [Beeferman et al., 1999, Christensen et al., 2005], lexical cohesion and conversational features [Galley et al., 2003]. [Galley et al., 2003] has compared the performance of the supervised

[29] Although semantic cohesion can exist in sentences that are intentionally coherent and intention coherence can exist without semantic cohesion, most sentences that relate coherently do exhibit cohesion as well.

classification approach that combines knowledge from various sources and the unsupervised lexical approach on predicting top-level boundaries in general meetings of the ICSI corpus. [Hsueh et al., 2006] has studied the problem of predicting topic boundaries at different levels of granularity and showed that the supervised classification approach performs better on predicting a coarser level of topic segmentation.

The task of topic labeling is complementary to that of topic segmentation. Prior research has proposed to model topics explicitly using generative models, in which a collection of mutually independent observations are probabilistically generated by a hidden topic variable [van Mulbregt et al., 1999, Schwartz et al., 2001, Blei and Moreno, 2001]. The generative topic models can also be used to hypothesize segment boundaries where the value of the topic variable for the next observation changes. In addition, research has also proposed to merge similar utterances into topic clusters using unsupervised clustering approaches that minimize inter-cluster similarity and maximize intra-cluster similarity [Ponte and Croft, 1997, Reynar, 1998, Choi et al., 2001, Utiyama and Isahara, 2001].

The Tasks As we would like to understand whether the finding on the preferred approach is generalizable to meeting structure centric or issue-based topic segmentations in the AMI corpus [Carletta et al., 2006], this study applies approaches previously proposed to segment topic boundaries in non-scenario meetings to the problem of identifying topic boundaries in the scenario-driven meetings of the AMI corpus. In this report, I report the results of the experiments of lexical chain-based segmentation and machine learning segmentation on meetings in the AMI corpus. In addition, as topic segmentation annotations in the AMI corpus come with a standardized set of labels, I also report the results of the topic labeling experiments, in which the task of automatic topic labeling is casted as multiple binary classification tasks

The task of topic labeling is a task complementary to that of topic segmentation. As topic segmentation annotations in the AMI corpus come with a standardized set of labels, this study casts the task of automatic topic labeling as multiple binary classification tasks: for each of the topics in the predetermined set, a classifier is trained to assign a hypothesized topic segment in the test set to either the class of belonging to this topic or not.

The Data Topic segmentations and labels have been annotated for the AMI meeting corpus collected across three sites, IDIAP, U of Edinburgh and TNO. Approximately two-thirds of AMI meetings are driven by a scenario, wherein four participants play the role of the project manager, marketing expert, industrial designer, and user interface designer in a design team, taking a design project from kick-off to completion. Annotators have the freedom to mark a topic as subordinated (down to two levels) wherever appropriate. As AMI meetings are scenario-driven, annotators are expected to find that most of the topics do recur. Therefore, they are given a standard set of topic descriptions that can be used as labels for each identified topic segment. Annotators will only add a new label if they cannot find a match in the standard set. The standard set of topic descriptions has been divided to three categories:

- Top-Level Topics refer to topics whose content largely reflects the meeting structure (e.g, presentation, discussion, evaluation, drawing exercise) and the key issues of the design task (e.g., project specs, user target group).
- Sub-Topics refer to parts of the top-level topics (e.g., project budget, look and usability, trend watching, components, materials and energy sources).
- Functional descriptions are those parts of the meeting that refer to either the varying process and flow of the meeting (e.g., opening, closing, agenda/equipment issues), or are simply irrelevant (e.g., chitchat).

In this study, we characterize a dialogue as a sequence of topical segments that may be further divided into subtopic segments. For example, the 25 minute meeting ES2012a, which aims at planning for a remote control design project, can be described by six major topics, from "opening" to "project specs and roles of participants" to "drawing exercise" to "project budget" to "discussion" to "closing". Depending on the complexity, each topic can

be further divided into a number of subtopics. For example, "discussion" can be subdivided to 2 subtopic segments, "existing products" and "look and usability". As this study is interested in comparing the segmentation algorithms on predicting topic boundaries at different levels of granularity, this research flattens the subtopic structure and consider only two levels of segmentation–top-level topics and all subtopics.

6.1.2 Method

Topic Segmentation In this study, I compare two segmentation approaches: (1) an unsupervised lexical cohesion-based algorithm (LCseg) using solely lexical cohesion information, and (2) a supervised classification approach that trains decision trees (C4.5) on a combination of lexical cohesion and conversational features.

The first approach, LCSeg, hypothesizes that a major topic shift is likely to occur where strong term repetitions start and end. The algorithm works with two adjacent analysis windows, each of a fixed size which is empirically determined. For each utterance boundary, LCSeg calculates a lexical cohesion score by computing the cosine similarity at the transition between the two windows. Low similarity indicates low lexical cohesion, and a sharp change in lexical cohesion score indicates a high probability of an actual topic boundary. The principal difference between LCSeg and TextTiling is that LCSeg measures similarity in terms of lexical chains (i.e., term repetitions), whereas TextTiling computes similarity using word counts.

The second approach employs the supervised classification framework, in which each potential topic boundary is labelled as either boundary (POS) or non-boundary (NEG). Our objective here is to train decision trees (c4.5) to learn the most predictive combinations of features that can characterize topic boundaries. This study uses the following features: (1) lexical cohesion features: the raw lexical cohesion score, the probability of topic shift indicated by the sharpness of change in lexical cohesion score and the prediction of LCSeg, and (2) conversational features: the number of cue phrases within 5 seconds preceding and following the potential boundary, similarity of speaker activity[30] within 5 seconds preceding and following each potential boundary, the amount of overlapping speech within 30 seconds following each potential boundary, and the amount of silence between speaker turns within 30 seconds preceding each potential boundary.

To ensure that various classifiers are compared at their optimal operating point, we propose perform a grid search for optimal parameter settings [van den Bosch, 2004]. The main goal of parameter search is to provide a fast approximation of parameter optimisation-by-validation. While small data sets do allow for some pseudo-exhaustive classifier wrapping, for larger data sets parameter search uses wrapped progressive sampling, a combination of classifier wrapping and progressive sampling. I first used features extracted with the optimal window size reported to perform best in Galley et al. (2003) for segmenting meeting transcripts into major topical units.

Topic Labeling As the topic labels in the AMI scenario-driven meetings are selected from a standardized set of topic descriptions, this study casts the task of automatic topic labeling as a task similar to text classification, in which each segment is assigned to appropriate classes given the transcript of speech. To ease the burden on classifiers, we convert the multi-class problem to multiple binary classification tasks: For each topic class, we compile the transcripts of speech in the segments that have been labeled as belonging to this topic class as its training data. Each segment is represented as a vector space of n-grams using the Benoulli model: a vector is 1 or 0 depending on whether the word appears in the segment. Then conditional Maximum Entropy (MaxEnt) models are trained from the training data to determine whether an unseen topic segment belongs to this topic class or not. The accuracy of these binary classification tasks will be used to assess the performance of this topic labeling approach.

The aim of this study is two-fold: first, we want to show whether it is possible to classify topics given only the lexical features extracted from the transcript. This is studied by examining the accumulated effect of all n-grams on topic classification accuracy. Second, we want to understand whether it is possible to attribute the classification accuracy to a subset of features that are indicative of the target topic class. To study this, we explore different feature selection criteria to identify the set of n-gram features that can characterize the target topic class.

[30] Measured as a change in probability distribution of number of words spoken by each speaker.

In particular, this study applies four measures, Log Likelihood (LL) and Chi-Squared (X2) statistics, Point-wise Mutual Information (PMI) and Dice Coefficient (DICE), to assess the "lexical discriminability" of each n-gram feature. Here, lexical discriminability is defined as the association strength between the occurrence of a given n-gram and that of a topic class. The hypothesis is that if an n-gram occurs significantly more often in a topic class than expected by chance, the n-gram is a discriminative feature of this topic class. All of these measures are calculated by first constructing a contingency table. The values 'a' and 'b' correspond to the observed frequency of a given n-gram ng in the target topic class $O(POS)$ and the observed frequency of that not in the topic class $O(NEG)$, whereas expected frequencies can be calculated as $E(POS) = c*(a+b)/(c+d)$ and $E(NEG) = d*(a+b)/(c+d)$. For the measure of LL and X2, we sum over terms of the form $log(O/E)$ and $(O-E/E)^2$. For the PMI and DICE measure, we compute the correlation between the observed frequency ($O(POS)$) of ng and the expected frequencies ($E(POS)$) if ng were independent.

The first category of power divergence family aggregates the significance test statistics, such as Pearsonś chi-squared (X2) and log likelihood (LL), into a single real-valued parameter. The parameter is computed by summing over the amount of variation between the observed frequency (O) and expected frequency (E) in each cell of the ngram contingency table. The second category of information theoretic measures conveys the mutual dependence of the occurrence of the ngram and the occurrence of decision-making subdialogues as a whole. Point-wise Mutual Information (PMI) and Dice Coefficient (DICE) are correlation coefficient for discrete events. In its original form, $DICE = 2P(X,Y)/(P(X)+P(Y))$. Here P(X), P(Y) and P(X,Y) are estimated by the occurrence count of the ngram (O(ngram)), occurrence count of all ngrams in decision-making subdialogues (O(in_decision)) and occurrence counts of the ngram in decision-making subdialogues (O(ngram, in_decision)) respectively.

	IN THIS TOPIC CLASS	¬ IN THIS TOPIC CLASS	TOTAL
TARGET N-GRAM	a	b	a+b
¬ TARGET N-GRAM	c-a	d-b	c+d-a-b
TOTAL	c	d	c+d

$$LL(ng) = \sum O \log(0/E) \tag{1}$$

$$X2(ng) = \sum((0-E)/E)^2 \tag{2}$$

$$PMI(ng) = \log(O(POS)/E(POS)) \tag{3}$$

$$DICE(ng) = 2O(POS)/(O(TOPIC)+O(ng)) \tag{4}$$

6.1.3 Evaluation

To evaluate the performance of segmentation models, we use metrics that have proven useful in the field of text segmentation (P_k). [Beeferman et al., 1999] has defined the P_k measure as the probability that a randomly drawn pair of utterances are incorrectly predicted as coming from the same segment. P_k was designed to overcome limitations in the use of precision and recall for text segmentation.

To evaluate the performance of topic classification models, classification accuracy as represented as f-score (F1) is calculated as follows. We loop over each topic in the standardized set and compute the precision and recall as the total number of segments that have been assigned correctly to the topic class divided by the total number of reference segments and hypothesized segments of the topic class respectively.

6.1.4 Results

Experiment 1: Predicting Top-level and Subtopic Segment Boundaries The scenario-driven meetings in the AMI corpus have on average eight top-level topic segments, which describe either the overall meeting structure and key issues of this product design, or serve the FUNCTIONAL purpose. In addition, the AMI meetings have

on average three more Sub-Topic segments that form parts of the Top-Level Topics. Compared to the ICSI corpus, AMI meetings are shorter, with relatively shallower hierarchies. [31] This experiment aims to explore whether approaches previously proposed to identify topic boundaries can be applied to identify meeting structure-centric topic-level topic boundaries and issue-based subtopic boundaries in the AMI scenario meetings.

In this experiment, we perform a five-fold cross validation. In each fold, we train models on 6 series of 4 meetings each and test on one unseen series of meetings. All of the results are reported on the test set. Table 34 shows the performance of the LCSeg and the feature-based classification models (CM) integrating the lexical cohesion and conversational features discussed in Section 6.1.2. Results show that CM performs considerably better than LCSeg on predicting topic boundaries in the scenario meetings. Also, CM performs better on the task of predicting meeting-structure centric boundaries at a coarser level than on the task of predicting boundaries of issue-based Sub-Topics at a finer level. The results are consistent with previous findings that feature-based approaches are preferred for finding coarser level topic boundaries. Results also suggest that the Sub-Topics annotated in the AMI corpus are similar to the top-level topics in the ICSI corpus in terms of the level of granularity. [32]

Error Rate (Pk)	LCSeg (k)	LCSeg (unk)	CM (c4.5)
ICSI (TOP)	25.77%	36.50%	28.35%
ICSI (SUB)	32.14%	32.31%	36.90%
AMI (TOP)	36.84%	42.49%	33.00%
AMI (SUB)	38.40%	41.31%	35.52%

Table 34: *Performance comparison of probabilistic segmentation models at the two levels of topic granularity: Top-Level Topics (TOP) and Sub-Topics (SUB).*

Experiment 2: Classifying Topics using only Lexical Features In this study, automatic topic labeling is casted as multiple binary classification tasks.Hence, we can investigate whether lexical features have distributions sufficiently different enough between topic classes by evaluating whether the models trained using only lexical features can yield reasonable accuracy on the classification tasks.[33]

Again, we performed a five-fold cross validation. Results suggest that simple lexical features can be used to automatically classify some of the topic classes, including some in the Functional category (e.g, "agenda", "closing") and those non-meeting structure centric, issue-based topics in the category of Top-Level Topics (e.g., "project spec", "user target group") and Sub-Topics (e.g., "budget", "component, materials and energy sources").

Experiment 3: Selecting Discriminative Features for the Task of Topic Classification Having established that it is possible to classify topic classes given the lexical features extracted from the transcript, we then investigate the question of whether a subset of lexical features can be identified as indicative of topic classes. The objective of this experiment is to identify feature selection methods (as discussed in Section 6.1.2) that can be used to further reduce the vector space of segment representations and improve the classification accuracy. To assess the effect of feature selection criteria on classification accuracy, we perform the following procedure: We first apply each of the four measures to calculate the lexical discriminability of all n-grams in the topic model and then sort n-gram features according to their computed lexical discriminability scores. Then we train classification models using the 25% most discriminative (Q1), the 25% mildly discriminative (Q2), the 25% mildly indiscriminative (Q3) and

[31] The AMI scenario meetings last around half an hour long, whereas the ICSI meetings last an hour in average. The meetings in the ICSI corpus have in average seven top-level topic segments and ten more subtopic segments.

[32] Note that because the procedure of obtaining the results on the ICSI and AMI corpus are different, the results reported here are not directly comparable across these two corpus.

[33] The set of n-gram used in training each topic classification model are obtained from the transcript of the segments of the given topic class in in entire AMI meeting corpus. For example, the language model trained to classify the topic class AGENDA includes 1103 uni-grams, 5824 bigrams, and 8223 tri-grams that have occurred in the 142 segments of Functional topic AGENDA.

Accuracy (F1)	1gram	1gram LL-Q1	1gram DICE-Q1
FUNCTIONAL (average)	0.57	0.62	0.54
FUNCTIONAL (Closing)	0.56	0.67	0.53
TOP-LEVEL (average)	0.45	0.53	0.48
TOP-LEVEL (Target Group)	0.36	0.63	0.38
SUB-TOPIC (average)	0.40	0.44	0.40
SUB-TOPIC (Budget)	0.50	0.71	0.57

Table 35: *Effect of n-gram features on the accuracy of classification models.*

the 25% least discriminative (Q4) of these sorted n-gram features. Finally, we examine the effect of features at different levels of lexical discriminability on classification accuracy.

We posit that if the lexical discriminability measure works well, the performance of the models trained using Q1 ought to outperform the models trained using other subsets of less discriminative features. Table 36 suggests that for models that are trained with the LL, DICE, or X2, Q1 features are better predictors of topic classification than Q2, Q3, and Q4 features. Results shown in Table 35 suggest that using discriminative features selected by the Log Likelihood ratio achieves the best performance in the task of topic classification, improving the average results of classifying Functional, Top-Level Topics and Sub-Topics by 9.4%, 27.6%, and 7.8% respectively. In other words, these lexical discriminability measures correspond well to the association strength between an n-gram feature and a topic class.

	Q1	Q2	Q3	Q4
LL	0.58	0.26	0.21	0.08
X2	0.51	0.39	0.41	0.08
DICE	0.55	0.24	0.00	0.00
PMI	0.00	0.09	0.28	0.29

Table 36: *Effect of feature selection methods on average classification accuracy (F1) of models trained with unigram features. Q1, Q2, Q3 and Q4 features refer to the most discriminative quarter, mildly discriminative quarter, mildly indiscriminative quarter, and least discriminative quarter of n-gram features.*

6.1.5 Outlook

In this study, we have quantitatively assess the effectiveness of approaches in both the task of topic segmentation and that of topic labeling. A natural next step is to develop probabilistic models that can perform these two tasks simultaneously. Also, we hope to incorporate contextual features by training conditional random fields. Finally, as we do not want to assume perfect human transcripts are always available, we will assess these models directly on automatic speech recognition output.

6.2 Topic Segmentation Using Conditional Random Fields

6.2.1 Introduction

In this section, we describe our experiments with automatic topic segmentation of AMI meeting transcripts. We propose a method based on Conditional Random Fields, n-gram classes, and Minimum Description Length-based discretization of numerical features. and report on a matrix of experiments. This section is organized as follows. In section 6.2.2 we motivate viewing topic segmentation as a sequential phenomenon, optimizing class probabilities in a global manner. In section 6.2.3, we briefly discuss previous work. Section 6.2.4 introduces Conditional random

Fields. In section 6.2.5 we describe the data at hand, our features and the representation of classes. In sections 6.2.6 and 6.2.7 we describe the experiments carried out and discuss our results. Finally, section 6.2.8 draws some conclusions and presents our plans for future work.

6.2.2 Topic Segmentation as a Sequential Phenomenon

Topic segmentation is essentially a wide-context phenomenon. The unit in which contexts are measured varies from sentences ([Choi, 2000]), blocks of sentences ([Galley et al., 2003],[Hearst, 1997]), and utterances ([Shriberg et al., 2000]).

Lexical cohesion-based algorithms, such as the well-known LCSEG ([Galley et al., 2003]), or its word frequency-based predecessor TextTile ([Hearst, 1997]) capture topic shifts by modeling the statistics of word repetition. This low-level type of information has proved to be quite valuable for topic segmentation, and as such is comparable to bag of words representations of documents for document classification: backing off from high-level descriptions of documents to low-level order-free representations leads to accurate classifiers. LCSEG works by shifting a fixed-size sentence-based window over a text, and computing the amount of overlap of lexical chains (word repetitions) with the left context of a candidate boundary, and its right context. An aggregate cosine quantity can be computed from these two quantities that expresses the amount of *cohesion* between the two contexts. If cohesion is low (below a boundary determined by e.g. standard t-test measures), a boundary is likely. Although LCSEG inherently is a contextual algorithm, the algorithm does not perform a non-local optimization of the sequence of assigned decisions. Yet, it has become clear in the past few years that topic segmentation, as any other sequential linguistic task, benefits from a global optimization of the sequence of assigned labels ('yes/no boundary') (see section 6.2.3 below.) Therefore, an interesting question to ask is whether LCSEG-based algorithms benefit from a Hidden Markov style optimization of the sequence of assigned decisions conditioned on features. CRFs are a natural environment to study this question. Therefore, we will use the raw cohesion scores produced by LCSEG in addition to a small number of other features to train CRFs.

6.2.3 Previous Work

Lexical cohesion-based models for topic segmentation have proved to be quite successful, cf. [Galley et al., 2003] and relatively robust against textual distortion (such as emanating from speech recognition). The idea of exploiting the sequential nature of topic labels for the purpose of topic segmentation is not new. [J. Yamron et al., 1998] were among the first to propose Hidden Markov techniques for topic segmentation, using the transition probabilities between topic labels and topic-specific language models. [Blei and Moreno, 2001] proposed an Aspect Hidden Markov Model using topic-specific language models for estimating emission probabilities and segment (cluster) models for estimating transition probabilities. Yet, the HMM based approaches all suffer from the limited expressiveness of the formalism: CRFs are much more expressive in being able to model unbounded dependenciesc across data, looking either forward or backward. In our work, we lift the limitation of limited contextual dependencies by switching to the much more expressive undirected graphical model structure of Conditional Random Fields.

6.2.4 Conditional Random Fields

Conditional random fields (CRFs) ([Lafferty et al., 2001]) have recently emerged as competitive algorithms in the machine learning community. CRFs are a probabilistic framework geared towards labeling sequential data. 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 optimization, 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. Normalization

over an entire state sequence leads to corrections of local errors. Given a undirected graph $G = (V,E)$, with V the set of vertices, 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 neighbors in the graph G. As defined in Lafferty *et al.* [Lafferty et al., 2001], the joint distribution of a label sequence $\mathbf{y}$ given an observation sequence $\mathbf{x}$ of features is

Definition 1

$$p_\theta(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \mathbf{y}_e, \mathbf{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathbf{y}_v, \mathbf{x}) \right)$$

The parameters λ, μ (feature weights) are estimated from the training data. The features f_k and g_k address edges (labels) and vertices (word level properties, like parts of speech) of the graph imposed on $\mathbf{y}$, respectively. Technically, they are functions returning 1 or 0 for the test they perform on the observation sequence. Features become conditioned on state (label) information; 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, making them a hybrid between MaxEnt and HMM. Viterbi-style best path algorithms can be used to produce the most likely state sequence explaining the observation sequence $\mathbf{x}$.

6.2.5 Data

The AMI meeting corpus consists of hand-annotated meetings from IDIAP, Edinburgh and TNO. Topical structure is annotated on the meeting speech transcripts with a granularity of two levels down: top level (main) topics and subtopics (two levels). For our experiments we used a 10-fold division of 35 meetings for training and testing, as agreed upon in WP5. Topic segmentation on the AMI data is a hard task: the vocabulary is technical and overlapping for many topics.

Data Representation and Features We have used spurt-based feature data provided by Edinburgh, in order to compare results. A spurt is consecutive speech without longer than .5 seconds of pauses between utterances. Not being delimited by the number of words, each spurt can be of varying length. Initial experiments showed that a wide context of spurts works better; therefore, we set the context width to 5 spurts to the left and 5 spurts to the right.

The following features are being used (these are also features used by [Galley et al., 2003]):

Feature	Description	Data type
1	Raw lexical cohesion score LCSEG	continuous
2	Cohesion probability	continuous
3	Number of cue phrases	continuous
4	Amount of overlap on the word level	continuous
5	Amount of gaps (pauses, silences)	continuous
6	Speaker activity change	continuous

Discretization Continuous features wil have to be discretized for CRF. We investigated two binning methods. The first (which we call 'FIXED'), 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. The second (called 'MDL') is a binning method based on the Minimum Description Length method of [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 Minimum Description Length principles is reached. We report results for both methods on the AMI data.

Class Space Expansion We decided to expand the class space from 2 (yes/no) to a larger class space. Recent work by [van den Bosch and Daelemans, 2005] 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:

- $(i-1)$ $f_1,\ldots,f_n,C_1$
- (i) $f_1,\ldots,f_n,C_2$
- $(i+1)$ $f_1,\ldots,f_n,C_3$

we can replace the class for instance i with $C_1+C_2+C_3$. 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 of instance i as well. Various possibilities now open up for combining these votes to produce the classification for instance i, such as majority voting, weighted voting, distance-based voting etc. Our intuition for CRFs is that sequence optimization works better for multiclass sequences than for binary sequences. Given the limitation of CRF++ to bigram 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 the 5-left-5-right spurt window one spurt. For the AMI data, on average, a number of 9 (2^3+1) classes was found after expansion. An example is the following class table (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 unigram 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 optimization strategy of CRFs. [34]

[34] An alternative is that, for a certain instance, we not only vote for classes on the basis of neighboring n-gram classes, but that we 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 (or 'records').

6.2.6 Experimental Setup

For our experiments, we used the 10-fold cross-validation partitionings agreed upon within WP5. For every 90-10% fold, we trained a first-order CRF on 90%, and tested on 10%, using Taku Kudo's CRF++ package ([Kudo, 2006]). This implementation has 3 hyperparameters:

- a `c` parameter controlling the tightness of fit to the training corpus
- an `f` parameter setting a count cut-off threshold for features (features with a frequency below the value for this parameter are ignored).
- an `e` parameter controlling the termination of the training process: if the difference between the log-likelihood of the previous and current parameter setting drops below the value of this parameter, training halts. Early stopping is a remedy against overfitting.

We fixed the `c` hyperparameter to `1.5` using grid search in the hyperparameter space for a separate held out development set based on ICSI data. The `f` hyperparameter was not optimized in order to guarantee full use of features. The termination criterion was to $error \leq 0.001$; this hyperparameter was not tuned as we did not observe any great difference between this value and the default value $e \leq 0.0001$. A fragment of our feature template (in the native CRF++ format) is listed below. It generates first-order (unigram) features, bigram and trigram features. Usually, when instantiated, the number of features per fold easily exceeds 10,000. The category model is a first-order Hidden Markov Model, $P(c_i \mid c_{i-1})$. Feature descriptions of instances are arranged in matrix form ($\%x[row, column]$), with $\%x[i, j]$ describing the j-th feature of instance i.

6.2.7 Results

We report results using the wellknown P_k and $WindowDiff$ error metrics ([Pevzner and Hearst, 2002]). While not uncontroversial, the use of these metrics is widespread.

Algorithm	Data	Average P_k	Average WindowDiff
LCSEG			
	SUB	38.4	38.6
	MAIN	37.4	37.6
CRF(FX)			
	SUB	36.3	38.4
	MAIN	31.2	34.2
CRF(MDL)			
	SUB	**33.2**	**35.2**
	MAIN	**27.9**	**32.1**

It appears that the MDL-based binning approach significantly outperforms the fixed-size binning method. Moreover, our approach is significantly better than LCSEG, presumably due to the use of sequential information. This hypothesis is yet to be validated, however, by deactivating the sequence optimization alltogether using a non-sequential Maximum Entropy classifier.

As we used 10-fold cross-validation, and Edinburgh 5-fold (which means we have used 10% more training data for every fold), results are not fully comparable between the two sites. Yet, it would seem that our method performs favorably. We plan to align our experiments with more scrutiny for the final deliverable this year.

6.2.8 Conclusions and Future Work

In this section, we evaluated a Conditional Random Field approach to topic segmentation of AMI meeting transcripts. Using spurt-based data, with features addressing lexical cohesion, topic cues, pauses and speaker activity, and expanding class space with n-gram classes, we were able to significantly outperform the LCSEG algorithm in a 10-fold cross-validation experiment.

We have generated a number of multimodal features not yet incorporated within our topic segmentation method. These features include:

- Visual activity cues (see the section on Hot Spot Segmentation): an estimate of the amount of motion per speaker during a meeting
- Speaker diarization: who's speaking when?
- Language modeling cues, such as mutual perplexity scores of language models built for left and right contexts of a candidate boundary
- Topic cues (term extraction)

We plan to insert these features into the spurt-based data we have been using for these experiments.

Further, we have implemented a genetic algorithm wrapper around LCSEG to search its hyperparameter space. Although we found an optimal setting of hyperparameters, these parameters were not used yet by Edinburgh to generate the LCSEG cohesion scores. Instead, these features were generated using the default parameter settings. For future experiments, we will use these parameter settings to optimize the performance of LCSEG, and hence boost the quality of our features.

To fully assess the gains from sequence optimization, we plan to perform additional experiments with non-sequential maximum entropy classification.

```
U110:%x[0,0]
U111:%x[0,1]
U112:%x[0,2]
U113:%x[0,3]
U114:%x[0,4]
U115:%x[0,5]
U116:%x[-4,0]
U116:%x[-4,1]
U116:%x[-4,2]
U116:%x[-4,3]
...
U120:%x[1,0]
U120:%x[1,1]
...
U120:%x[1,4]
U120:%x[1,5]
U121:%x[2,0]
U121:%x[2,1]
U121:%x[2,2]
...
U122:%x[3,2]
...
U123:%x[4,1]
U123:%x[4,2]
U123:%x[4,3]
U123:%x[4,4]
U123:%x[4,5]

U124:%x[-4,0]/%x[0,0]
U124:%x[-4,1]/%x[0,1]
U124:%x[-4,2]/%x[0,2]
...
U132:%x[-4,0]/%x[0,0]/%x[4,0]
U132:%x[-4,1]/%x[0,1]/%x[4,1]
U132:%x[-4,2]/%x[0,2]/%x[4,2]
....
```

Figure 14: Fragment of CRF feature template.

7 Addressing

7.1 Introduction

When a speaker contributes to a conversation, he does not only perform a certain communicative act but he also performs that act towards other participants. This form of orientation and directionality of the act the speaker performs is referred to as addressing. In a conversation involving two participants, the hearer is almost always the addressee of the speech act that the speaker performs; the speaker may also talk to himself. However, in a multi-party conversation, a speaker may address his utterance to a single participant, to a subgroup of participants or to the group as a whole.

In the AMI project, we aim to develop a meeting browser that will provide users with relevant information about meetings. For answering questions such as "Who was asked to prepare a presentation for the next meeting?" or "Were there any arguments between participants A and B?" some sort of understanding of the dialogue structure is required. For inferring such a structure, it is not enough only to identify conversational acts that participants perform, but also to identify the addressees of those acts.

In this section, we present results on addressee classification on the AMI data using several static Bayesian Network (BN) classifiers. As addressing is carried out through various communication channels, we made use of different multimodal features obtained from gaze, speech and contextual resources. The results presented in this section are obtained using hand-annotated features.

Classification Task In a dialogue situation, which is an event that lasts as long as the dialogue act performed by the speaker in that situation, the class variable is the addressee of the dialogue act (**ADD**). We define the addressee classifier to identify for each *dialogue act* that is not labelled as BACKCHANNEL, FRAGMENT, STALL or OTHER whether the act is addressed to a particular individual (**A**, **B**, **C** or **D**) or to a group (**Group**).

Since the AMI addressee annotation schema does not make a distinction between addressing a subgroup of participants and addressing the group as a whole, the Group label is used to mark any group of participants as addressee [AMI, 2005]. Furthermore, the schema allows for dialogue acts to be labelled with the Unclassifiable addressee tag which denotes that annotators could not determine who is being addressed. These instances of dialogue acts were employed for deriving contextual information used for predicting the addressee of the dialogue act at hand.

Data Sets Only a small part of the AMI scenario-driven collection has been annotated with addressee information. For the experiments presented in this section, we selected 13 meetings that were annotated with addressees and focus of attention: ES2008a, TS3005a, IS1000a, IS1001a, IS1001b, IS1001c, IS1003b, IS1003d, IS1006b, IS1008a, IS1008b, IS1008c, IS1008d. Although IS1006d was also annotated with addressees and focus of attention, we excluded this meeting from our data set because it was not annotated with all types of information that we used in some of our experiments. In the selected data set, approximately 56% of instances are addressed to a group whereas the remaining instances are almost equally distributed over individual addressee values.

7.2 Methods

Addressee classification on the AMI data was performed by means of several static BN classifiers using various Bayesian Network classifier learning algorithms implemented in WEKA [Witten and Frank, 2000]. We experimented with the Naive Bayes (NB), Tree-Augmented Naive Bayes (TAN), BN augmented Naive Bayes (BAN) and general Bayesian Network (GBN) classifers. These classifiers differ based on the structures that are permitted. NB and augmented NB classifiers treat a classification node as a special node which is defined as a parent of all feature nodes. In NB classifiers, feature nodes are not linked; features are conditionally independent given the class variable. TAN classifiers extend NB classifiers by allowing feature nodes to form a tree structure. Similarly, BAN classifiers extend NB classifiers by allowing feature nodes to form an arbitrary graph. Unlike (augmented) NB classifiers, GBN classifiers treat the classification node as an ordinary node that can be linked to an arbitrary

number of feature nodes. For an overview of the these classifiers, we refer to [Cheng and Greiner, 1999]. Regarding the BAN classifier, we experimented with several thresholds for the maximal number of parents for each node. As the accuracies obtained for different parent thresholds were nearly identical, we report here the classification results for the best performing BAN classifier.

The K2 algorithm was applied for learning the structure of the BAN and GBN classifiers [Cooper and Herskovits, 1992]. The structure of the TAN classifier was learned using the algorithm presented in [Friedman et al., 1997]. For learning parameters of the static BN classifiers we used the algorithm implemented in WEKA that produces direct estimates of the conditional probabilities.

7.3 Evaluation

For evaluating performances of the addressee classifiers, we performed stratified 10-fold cross validation using the whole data set. It assumes that in each fold the class is represented in approximately the same proportions as in the full data set. The division of the data set into ten folds in a way that satisfies the stratification criterion is done automatically in WEKA. In addition to overall accuracy, the detailed accuracies per class value have been estimated in terms of precision, recall and F-measure.

7.4 Previous Work - Summary of Findings

In our previous work on addressee identification in face-to-face meetings, we focused on the exploration of the appropriate feature types and models for this type of task[35]. The experiments were conducted on the M4 corpus using the NB classifier and the BAN classifier, which was chosen as a representative of more general Bayesian Network classifiers. The M4 corpus is described in detail in [Jovanovic et al., 2005]. It consists of short discussion meetings that are scripted in terms of type and schedule of meeting actions, but content is natural and unconstrained.

In summary, the following conclusions were drawn from the experiments on the M4 data:

- Contextual information aids classifiers' performances over gaze and utterance information.
- Utterance features are the most unreliable cues for addressee prediction.
- Listeners' gaze direction provides useful information only in the situation where gaze features are used alone.
- Combinations of features from various resources improve classifiers' performances in comparison to performances obtained from each resource separately.
- The highest accuracies are achieved by combining contextual and utterance features with speaker's gaze directional cues.
- The BAN classifier outperforms the NB classifier over all feature sets, although the difference is significant only when contextual features are exploited.

7.5 Results

To compare results on addressee classification on the M4 and AMI data, we explored how well the addressee of a dialogue act can be predicted on the AMI data using a set of features that was shown to lead to the highest accuracies of the addressee classifiers on the M4 data (henceforth, *M4 feature set*). Then, we investigated whether the performances of addressee classifiers on the AMI data can be improved using a modified set of contextual, utterance and gaze features (henceforth, *AMI feature set*). Additionally, the impact of meeting context on the classifiers' performances was explored.

[35]The work was reported in the AMI deliverable D5.1 http://research.amiproject.org/

7.5.1 Addressee Classification using Static BN Classifiers - M4 Feature Set

We estimated performances of the BN classifiers on the AMI data using the feature set that was shown to score the highest accuracy on the M4 data. The set contains the following contextual, utterance and gaze features:

- **Contextual features**
  - **SP-R**, **ADD-R** and **DA-R** - the speaker, the addressee and the dialogue act type of the related dialogue act. A dialogue act in the AMI schema can be either unrelated or related to a previous dialogue act produced by the same or by the different speaker or to something that is not expressed verbally. The contextual feature set encompasses only those related dialogue acts that are produced by a different speaker. In all other cases, the dialogue act is considered as unrelated.
  - **SP-1**, **ADD-1**, and **DA-1** - the speaker, the addressee and the dialogue act type of the immediately preceding dialogue act on the same or a different channel.
  - **SP** - the speaker of the current DA
- **Utterance features**
  - **PP** - does the utterance contain personal pronouns "we" or "you", both of them, or neither of them?
  - **PPA** - does the utterance contain possessive pronouns or possessive adjectives ("your/yours" or "our/ours"), their combination or neither of them?
  - **IP** - does the utterance contain indefinite pronouns such as "somebody", "someone", "anybody", "anyone", "everybody" or "everyone"?
  - **Name-X** does the utterance contain the name of participant `X` where $\texttt{X} \in \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$?
  - **DA-Type**- the type of the current dialogue act
  - **Short**- whether or not the utterance duration is less or equal to 1 sec.
- **Gaze features**
  - **SP-looks-X**, **SP-looks-NT** - the number of times the speaker looks at participant X or looks away (NT) during the time span of the utterance: `zero` for 0, `one` for 1, `two` for 2 and `more` for 3 or more times; $\texttt{X} \in \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$

The features exploited for the experiments on the AMI data differ from the original feature set in the value set for the DA feature and in line with that for the DA-1 and DA-R features: the M4 and AMI meetings were annotated using different dialogue act tag sets. The AMI dialogue act tag set is described in Section 2.1.4. The M4 meetings were annotated with a dialogue act tag set that is based on the MRDA (Meeting Recorder Dialogue Act) tag set [Dhillon et al., 2004]. Similar to the experiments on the M4 data, we used the complete set of *relevant dialogue act tags* as the value set for the **DA** feature on the AMI data. Irrelevant dialogue act types for addressee prediction on the AMI data are BACKCHANNEL, STALL, FRAGMENT and OTHER.

Table 37 summarizes the accuracies of the BN classifiers estimated on the M4 and AMI data using the M4 feature set.

Data	NB	TAN	BAN	GBN
AMI	74.01%	76.23%	76.69%	76.62%
M4	78.49%		82.59%	

Table 37: Accuracies of the BN classifiers estimated on the AMI and M4 data using the M4 feature set

The results show that accuracies for different classifiers do not differ significantly on the AMI data, although more general BN classifiers outperform the NB classifier. The results also indicate a significant decrease in the

performances of the NB and BAN classifiers on the AMI data in comparison to their performances on the M4 data. As shown in Table 38, both classifiers perform similarly on both data sets regarding the group classification. However, the classifiers perform significantly worse on the AMI data regarding the individual addressee values.

	NB		**BAN**	
Class	AMI	M4	AMI	M4
Group	0.798	0.796	0.811	0.826
A	0.647	0.770	0.711	0.802
B	0.674	0.789	0.715	0.840
C	0.629	0.791	0.692	0.826
D	0.667	0.746	0.721	0.834

Table 38: F-measures per class value for the NB and BAN classifiers on the M4 and AMI data sets

The investigation of the performances of classifiers obtained using each feature type separately as well as using various combinations of feature types have shown that the selected utterance and gaze features are less effective cues for prediction on the AMI data than on the M4 data. For example, when using solely gaze features, the BAN classifier scores considerably lower F-measures for individual addressee values on the AMI data (mean F-measure=0.361) than on the M4 data (mean F-measure=0.591). This drop of effectiveness of the gaze features on the AMI data regarding identification of single addressed dialogue acts can be influenced, among other things, by the presence of additional attention distracters in the meeting room e.g. remote control prototypes, laptops etc.

7.5.2 Addressee Classification using Static BN Classifiers - AMI Feature Set

We defined a modified set of utterance, gaze and contextual features focusing mostly on the better exploitation of contextual information as it was shown that conversational context contributes to the greatest extent to addressee prediction. Additionally, experiments were conducted with an extended feature set including a feature that conveys information about meeting context.

Contextual Features As to conversational context, we experimented with two notions of context regarding the preceding dialogue acts: local and global. In both cases, only relevant dialogue acts were taken into account. As defined above, irrelevant dialogue acts are those dialogue acts that are marked as BACKCHANNEL, STALL, FRAGMENT or OTHER.

The local context encompasses contextual information obtained from the relevant dialogue acts from the same or 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-grams), we also experimented with the extended contexts that includes 2 (2-grams) and 3 (3-grams) 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. Furthermore, we experimented with different *window-sizes* regarding the number of preceding turns as well as regarding the number of preceding dialogue acts within the same turn. 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. We conducted a number of experiments with various windows-sizes regarding the preceding turns as well as regarding the preceding dialogue acts within the same turn. The highest accuracies reported in this section were achieved using the contextual information provided from 2 preceding turns (**SP-T-1**, **ADD-T-1**, **DA-T-1**, **SP-T-2**, **ADD-T-2**, **DA-T-2**) and from the immediately preceding dialogue act within the same turn (**ADD-1**, **DA-1**). Furthermore, the preceding turns of the current speaker were also taken into account.

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 conversational feature set when experimenting with the local context features as well when experimenting with the global context features.

Gaze Features **SP-looks-X**, **SP-looks-NT** - whether or not the speaker looks at participant X or whether or not he looks away during the time span of the current dialogue act; $\texttt{X} \in \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$

Utterance Features Using the available annotations of dialogue acts, reflexivity and named entities, we experimented with a variety of utterance features that are considered with the content and the conversational function of the current dialogue act. However, the following features were shown to be the most informative when combined with gaze and contextual features:

- **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".
- **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.
- **DA-Type** - {inform, assess, social, elicit, offer, suggest, comment-about-understanding}. All elicitation classes (e.g. elicit-inform or elicit-assess) have been grouped into the *elicit* category; both social acts be-positive and be-negative have been grouped into the *social* category.

Meeting Context The meeting context is modelled in terms of the **Topic** feature. Although the AMI topic segmentation schema allows topics to be nested to several levels, we experimented only with *top-level* topics [Xu et al., 2005]. Top-level topics reflect largely the meeting structures based on the meeting scenario. 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*, *A-present*, *B-present*, *C-present*, *D-present*, *discussion*, *prototype presentation*, *prototype evaluation*, *project evaluation*, *costing*, *drawing*, *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 *A-present*, *B-present*, *C-present* and *D-present* incorporating in that way the background knowledge of the participant roles into the classification models.

Table 39 summarizes the accuracies of the BN classifiers obtained using the gaze and utterance features combined with the contextual feature set that contains local context features.

Context	NB	TAN	BAN	GBN
1-grams	73.70%	76.67%	77.04%	76.62%
2-grams	76.08%	78.20%	78.44%	77.51%
3-grams	77.43%	77.68%	77.36%	77.32%

Table 39: Accuracies of the static BN classifiers using the AMI feature set - local context

The comparison of the accuracies of the BN classifiers for the 1-grams model and the accuracies obtained using the M4-feature set shows that using the simplified set of utterance and gaze features as defined in the AMI feature set leads to the similar performances of all BN classifiers as when using the M4 feature set. The results indicate that when extending the local context, i.e. when going from the 1-grams model to the 2-grams model all classifiers

show improvements in the performances, although the NB classifier gains the most (about 2.5%). However, the further extension of the local context leads to the decrease in the accuracies for all classifiers except for NB. The results also show that for the 1-grams and 2-grams models the augmented NB classifiers and the GBN classifier outperform the NB classifier whereas for the 3-grams model all classifiers show similar performances. However, the highest accuracies are achieved for the BAN classifier using the utterance and gaze features combined with the contextual feature set that includes information obtained from 2 preceding dialogue acts. The confusion matrix and the detailed accuracies per class value for the BAN classifier are presented in Tables 40 and 41. As shown in the confusion matrix, most misclassifications were between individual and group addressing.

	Precision	Recall	F-measure
A	0.712	0.767	0.738
B	0.717	0.745	0.731
C	0.706	0.734	0.720
D	0.732	0.763	0.747
Group	0.842	0.809	0.825

Table 40: Evaluation per class value for the BAN classifier using utterance, gaze and conversational context features - 2-grams

	A	B	C	D	G
A	513	8	12	11	125
B	6	403	13	9	110
C	4	17	425	13	120
D	12	4	15	431	103
G	186	130	137	125	2448

Table 41: Confusion matrix for the BAN classifier; rows - actual values; columns - classified values; G - Group

Table 42 summarizes the accuracies of the BN classifiers using (1) the contextual feature set which includes global context features combined with utterance and gaze features and (2) the extended feature set encompassing the meeting context feature.

Feature set	NB	TAN	BAN	GBN
C+U+G	76.37%	78.51%	78.79%	77.64%
C+U+G+T	76.84%	78.96%	79.29%	77.49%

Table 42: Accuracies of the static BN classifiers using the AMI feature set: C - context features (global context) , U - utterance features, G - gaze features, T - topic feature

The results presented in the row C+U+G show that modelling conversational context in a way to include information obtained from previous turns as well as from the turn in progress does not improve significantly classifier performances in comparison to performances obtained using the local 2-grams context (see Table 39). Furthermore, the results indicate that the NB and augmented NB classifiers show little gain from the information about meeting context (about 0.5%) whereas the GBN classifier scores slightly lower accuracies when the meeting context information is employed.

7.6 Outlook

In this section we compared the performances of several static BN classifiers for the task of addressee prediction on the AMI data using hand-annotated features obtained from gaze, utterance and contextual resources. As the contextual feature set includes information about the addressees of the preceding dialogue acts, we are currently exploring how well the addressee of the current dialogue act can be identified using the predicted instead of the gold standard value for the addressees of the previous dialogue acts. For that purpose, DBN classifiers are being employed. This can be seen as the first step towards the automation of addressee identification on the AMI data.

8 Argumentation

8.1 Introduction

This section describes an approach able to capture the lines of the deliberated arguments in meeting discussions. This approach, the TAS-schema, was introduced in [Rienks and Heylen, 2005] and promises to be a valuable technique for capturing organizational memory. The structure that the argument trees encapsulate reveals information about the trail or path that has been taken in a meeting. It shows the line of reasoning at specific moments in time. The method can aid querying and summarization systems and is being used in meeting browsers (See fig 15). The possibility of preserving the arguments and their coherence relations for future explorations make them potentially valuable documents that contain a tacit representation of otherwise volatile knowledge [Shum, 1997, Pallotta et al., 2005].

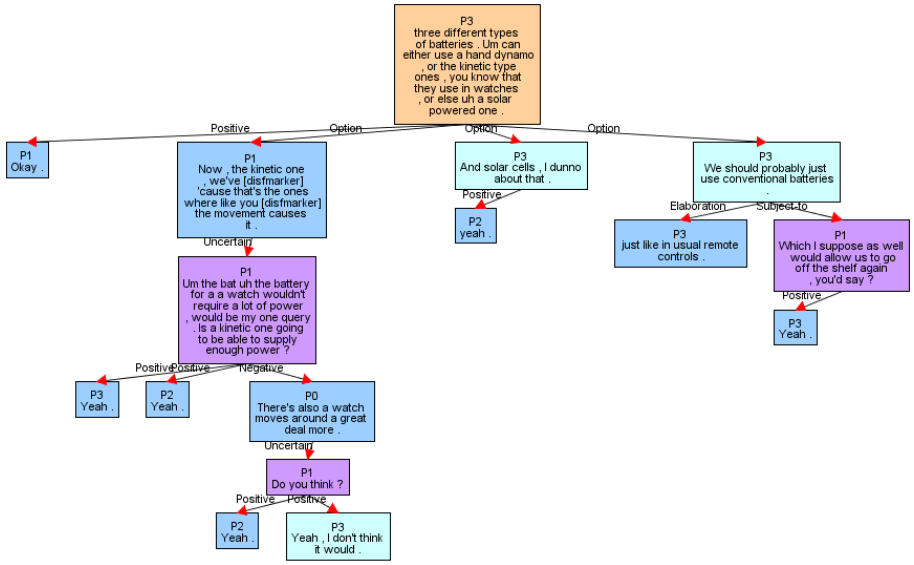

Figure 15: A typical TAS Argument Diagrams.

For end users of the representations, argument diagrams constitute a representation of the content of a conversation that leads to quicker comprehension, deeper understanding. They enhance the ability to detect weaknesses or flaws in the argumentation [Schum and Martin, 1982, Kanselaar et al., 2003]. Furthermore is has been claimed that they aid the decision making process and that they can be used as an interface for communication to maintain focus, prevent redundant information and to save time [Yoshimi, 2004, Veerman, 2000].

Here we present initial research efforts in this area. Eventually we aim to build a system that can automatically detect discussion segments, tag individual contributions with TAS-unit-labels, depict and label the relations between the units using the TAS-relation-labels and generate a visualization of the argument diagram. The complete label set is shown in Table 43.

See *http://hmi.ewi.utwente.nl/video4ami/UT_argumentation.wmv* for a video about the TAS-schema and its applications. An example of a TAS argument diagram, embedded in a meeting browser application, is shown in Figure 16.

Node labels	Relation labels
Statement	Positive
Weak statement	Negative
Open issue	Uncertain
A/B issue	Request
Yes/No issue	Specialization
	Elaboration
	Option
	Option exclusion
	Subject-to

Table 43: The labels of the Twente Argument Schema

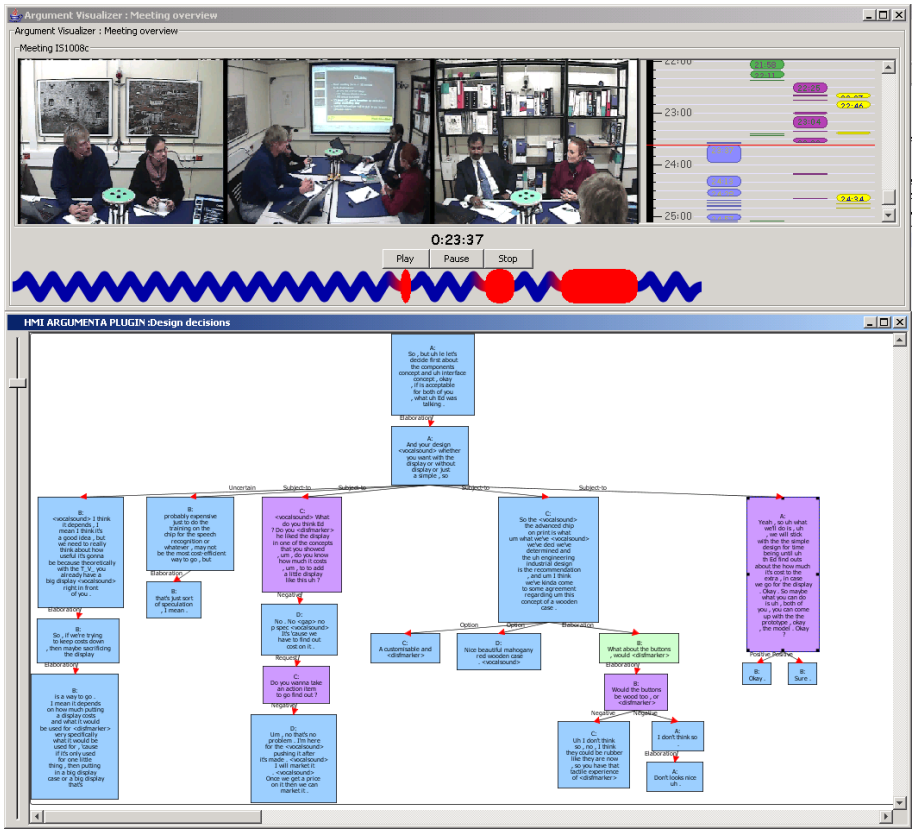

Figure 16: TAS Argument Diagrams in use as part of a meeting browser.

8.2 Creating a Corpus of Meeting Discussions

TAS was initially devised to create argumentation diagrams for AMI meetings. To perform the manual TAS annotations, the annotation tool *ArgumentA* was designed by using a number of components described in [Reidsma et al., 2005]. ArgumentA allows annotators to select text on a transcription-view pane and label them.

The label is assigned by selecting the unit text with the mouse from the transcription pane and then pressing a button that makes a label selection window pop-up from which the unit label can be picked. The labelled units appear on a canvas where they can be attached to the graph via an intuitive drag and drop interface. Once attached, a popup window appears from which the relation-label can be chosen. The resulting trees can be saved in different XML formats.

Three annotators were trained in several iterations. Apart from collectively developing the schema, elaborate discussions were held after a number of training sessions about when and why to pick a particular label in that particular case. The corpus, as it stands, comprises a total of 256 annotated discussions (diagrams) including over 5000 unit labels and 5000 relation labels.

8.3 Reliability of the TAS Schema

With respect to the issue of reliability one should first note that it is very well possible to end up with several diagrams from one discussion as there are likely to be more than one possible interpretation. [Walton, 1996] for instance showed that various different argument diagrams can be instantiated by one single text. Moreover, in Rhetorical Structure Theory (RST) [Mann and Thompson, 1988], which addresses similar issues as the TAS scheme, the suggestion is made that the analyst should make *plausibility judgements* rather than absolute analytical decisions, implying that more than one reasonable analysis may exist.

To measure the reliability of the scheme we therefore compared the unit labels on pre-segmented discussions for four meetings (12 discussions) between two annotators. The reliability issue for the relation part of the scheme is still under investigation. It turned out that, especially in first trials the value of Cohen's kappa (κ) [Cohen, 1960] were rather low (0.50) as a lot of confusion existed amongst the labels 'other' and 'statement'. This was resolved by a consensus definition, after which κ rose to a more acceptable value (0.87).

We also experimented with other ways to obtain reliability score based on more data. We applied techniques comparable to those introduced in [Steidl et al., 2005], by setting out the results of a classifier trained on (unit label) annotations of one annotator against the values provided by another annotator. (See Section 8.4).

8.4 Classification of TAS-unit Labels

In this section we report on our first experiments related to the automatic classification of the TAS unit labels.

8.4.1 Features

Except for the *lastlabel* feature, we only used lexical features.

? and OR A good indicator for an issue is a question mark. The *?-feature* gives a binary value whether a question mark is present or not. If a question mark is available, the number of times the word *or* appears is counted and used as a feature. (If the classification is based on transcripts derived from automatic ASR, a substitute for the question mark feature is needed.)

Length The length (number of words) of each segment is a feature. This feature helps to make a distinction between the *statement* and *other* labels.

Last Label Since discussions have the property of having some coherence we might expect that given the label of a segment the conditional chance of the label of the next segment might differ from the unconditional chance. Therefore the *lastlabel* feature, which is a bigram of the previous two labels, is used.

N-gram Points The n-gram-point feature is used to reduce the number of features. At first, all bi-, tri- and quadri-grams are computed for all segments. Then, for each label a predictivity score is computed and the X most predictive n-grams are selected. The predictivity score is equal to the product of the times the ngram occurs in nodes labeled X *and* the part of this 'ngram-space' occupied by nodes of type X. For example, the score for the ngram 'what do you' (see table 44) for type *statement* is $\frac{3}{3+0+100+97+2+0} \times 3 = 0.045$.

Using the ngrams selected points, an utterance is assigned ngram points by computing all ngrams in an utterance and enumerating all the occurrences of all ngrams per order and label. If for example the trigrams listed in Table 44 are found in an utterance and the occurrences of the ngrams in the training set are as shown in the table, than this utterance will get 69 points for the *statement - trigram* feature, 31 for the *weak statement - trigram* feature and so on.

POS-ngram Points The POS n-gram-point features are quite similar to the n-gram point features. But instead of attributing points to words, points are attributed to n-grams of Part-of-Speech tags.

trigram	statement	weak statement	open issue	a/b issue	y/n issue	unknown
what do you	3	0	100	97	2	0
do you think	3	1	97	92	100	0
we have to	63	30	50	1	93	4

Table 44: Examples of a trigrams found in an utterance and available in the training set

Perl scripts were used to extract the features *? and OR*, *Length*, and *Last Label* from our XML-format. The construction of n-grams was done using the N-gram Statistic Package (NSP) [Banerjee and Pedersen, 2003]. Using the Stanford Part-of-Speech tagger all segments were tagged to make POS-n-gramming possible [Toutanova et al., 2003].

8.4.2 Baseline

The corpus as it stands is unbalanced, consisting of 4245 *statements*, 199 *weak statements*, 244 *open issues*, 72 *a/b issues*, 460 *yes/no issues* and 3061 *others*. As a baseline we have used the implementation of a one-rule classifier resulting in a correct score of 69.1%. To see how our features would perform on a balanced corpus we also constructed a balanced corpus, having an equal number of nodes for each unit type. The baseline was again computed using a one-rule classifier, which resulted in an accuracy of 28.33%.

8.4.3 Results

We tried out different Machine learning techniques to produce our results, but looked into most detail at Weka's **J48** implementation of the C4.5 decision tree algorithm [Quinlan, 1993], since this classifier gave the best results as a baseline classifier compared to seven other classifiers available in Weka. Furthermore Weka's **DecisionTable** and **MultilayerPerceptron** were used on our most promising results. All our results were obtained after a 10 fold cross-validation. Here we only present our best results, a more extensive presentation of experiments and results can be found in [Verbree, 2006].

Our best result on the unbalanced corpus is 78.52% which shows an improvement of 9.4% on the best baseline. The combined confusion matrix produced by the J48, (Table 45) shows that improvement could be obtained by features that distinguish between utterances with the label *statement* or *unknown*. The table also shows that a label such as *ab_issue* is often incorrectly classified as it has only few occurrences.

a	b	c	d	e	f	$<--$ classified as
19	15	22	1	0	15	a = ab_issue
7	116	47	9	0	65	b = open_issue
8	31	3722	388	36	60	c = statement
1	9	668	2365	2	16	d = unknown
0	2	162	21	11	3	e = weak_statement
15	45	121	9	1	269	f = yn_issue
header						

Table 45: Confusion matrix of unbalanced J48-classifier on our best result

On the balanced corpora our best result was 51.43% which shows an improvement of 23.1% on the best baseline.

8.4.4 Elaborating on the Reliability Issue

In section 8.3 κ-measures were computed for the TAS annotation of the HUB corpus. Two problems met there were the small amount of discussions that could be compared and the absence of utterances of type *A/B issue* in each annotation. To get more insight in the reliability of our corpus we performed experiments where the J48 classifier was trained using parts of the corpus annotated by one annotator (row) and was tested on a part of the corpus annotated by another annotator (column). This *'virtual kappa'* overcomes the issue of sparse data and provides interesting views on the annotations. This resulted in the performances shown in table 46. When both training and test sets were picked from the same annotator, we used 10-fold cross-validation.

Trained / Tested on	Annotator 1	Annotator 2	Annotator 3
Annotator 1	84.4%	75.7%	70.3%
Annotator 2	75.6%	79.5%	66.2%
Annotator 3	67.0%	66.2%	82.2%

Table 46: Performance amongst annotators

We see that annotator one and two annotated in a much more similar way than annotator three in relation to the other two. This implies that the actual performance of the classifier can be even higher in the case that the annotators agree more strongly.

8.5 Discussion and Future Work

8.5.1 Relation with DA-Tagging

The classification task described in this section is very similar to dialog-act tagging. Research in this field mostly concentrates on cues that are either manually [Hirschberg and Litman, 1993] or automatically [Reithinger and Klesen, 1997] selected. The biggest difference for our approach in comparison to earlier dialogue act classifying approaches is the use of an ngram selection method. This method selects the most predictive ngrams from the total set of ngrams acquired. We have also experimented with *compressed* feature sets. The compression decreases the size of our feature vector and therefore also decreases our computing time. This of course, by itself not an advantage, unless we maintain accuracy. In addition to the compression, we also made use of n-grams of POS-tags as has previously been done in research on the generation of backchannels in a spoken dialogue system [Cathcart et al., 2003]. Using the same ngrams an accuracy of 78.52% was obtained without making use of compression and a result 77.20% when using compression. These results are based on the use of the J48 classifier.

8.5.2 Research on other ngram-selecting Methods

Our work has mostly concentrated on ngrams of words an POS-tags. Results of the experiments show that for each classifier the ngram-selecting method strongly influences the performance. More research on scoring algorithms might results in better ngram selection methods and therefore a better performance on the classification task. It is not just the selection of the right ngrams that influences the performance of our classifiers based on ngrams, however. Also the points attributed to a feature when a ngram is present are important. In our study we have used the number of occurrences of an ngram as a feature value. It might be worth the effort to research other possible values one could assign to an ngram.

8.5.3 Researching the Punctuation Features

The use of the presence or absence of a question mark as a feature could be regarded as a form of 'cheating', since in automatic speech recognition it is very hard to recognize wether an utterance is a question or not and thus deciding on placing a question mark in the output or not (See e.g. [Huang and Zweig, 2002]). Since we like to have or classification of a discussion using TAS to be applicable to discussion transcribed using automatic speech recognition we are considering the omission of this particular feature. Ongoing work investigates the influence of the *? and or* feature on the performance.

8.5.4 Application in JFerret

A plug-in has been developed for the JFerret meeting browser [Wellner et al., 2004]. Users are able to access the discussions depicted on a meeting time line. For each discussion the resulting argument diagram appears allows a quick grasp of the content of the on-going discussion. Clicking on the nodes in the diagram shifts the browser directly towards the corresponding moment in the meeting.

Eventually the possible applications for meetings annotated with the TAS schema are endless. They can be used for automatic summarization purposes, or aid processes aiming to find out who adhered to a specific opinion at any given moment. They can be used to see who proposed the accepted solution, or who objected to most of the discussed points. Managers can use the diagrams to investigate what went well or wrong in the discussion and which arguments were made in favor or against a specific proposal. For more information about the sorts of applications we foresee to emerge refer to [Rienks et al., 2006].

8.5.5 Future Work

There are currently three lines of research that we are engaged in with respect to the Argumentation Schema.

Up till now we have focused on node classification only. We are currently working on relation classification as well. Our first approach to the classification of relations are discussed in [Verbree, 2006].

In the end, the system we would like to have the system work in real time. We are therefore considering to run tests directly on the ASR output.

Finally, investigations have started to measure the actual benefit of the use of argument diagrams in a meeting browser. Does presenting a Argument Diagram really improve the system? (i.e. are user queries answered quicker with a higher satisfaction rate?) This is certainly an important topic [Van Gelder, 2001].

8.6 Conclusions

We made some initial, but important steps on the road to automatic generation of argument diagrams. A corpus containing over 250 argument diagrams was created. Machine learning experiments on automatic tagging the unit-labels resulted in a performance of 78.52% on our unbalanced and an average of 51.43% on our balanced test set using a J48 classifier.

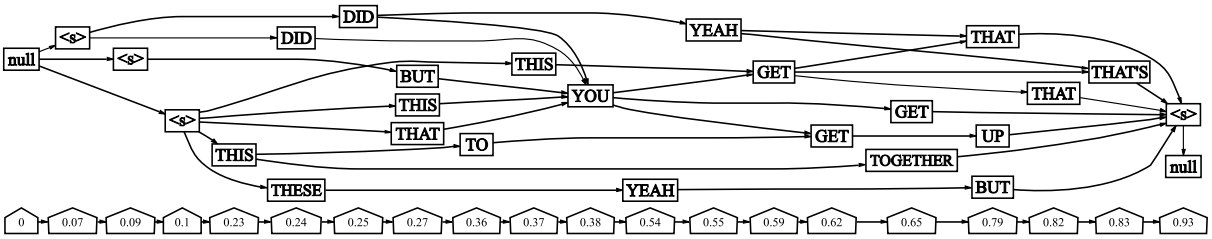

Figure 17: Example of a word lattice

9 Speech indexing and retrieval

9.1 Introduction

This section presents two approaches to spoken document retrieval – search in LVCSR recognition lattices and in phoneme lattices. For the former one, an efficient method of indexing and search of multi-word queries is discussed. In phonetic search, the indexation of tri-phoneme sequences is investigated. The results in terms of response time to single and multi-word queries are evaluated on ICSI meeting database.

Unlike search in text, where the indexing and search is the only "science", spoken document retrireval (SDR) is a more complex process that needs to address the following points:

- conversion of speech to discrete symbols that can be indexed and searched – large vocabulary continuous speech systems (LVCSR) and phoneme recognizers are used. Using phoneme recognizer allows to deal with out-of-vocabulary words (OOVs) that can not be handled by LVCSR.
- accounting for inherent errors of LVCSR and phoneme recognizer – this is usually solved by storing and searching in word, respectively phoneme lattices (Fig. 17) instead of 1-best output.
- determining the confidence of a query – in this paper done by evaluating the likelihood ratio between the path with searched keyword(s) and the optimal path in the lattice.
- processing multi-word queries, both quoted (exact sequences of words) and unquoted.
- providing an efficient and fast mechanism to obtain the search results in reasonable time even for huge amounts of data.

In this section, we do not deal with pre-processors such as LVCSR system and phoneme recognizer, but concentrate on indexing and search issues. Section 9.2 reviews the LVCSR-based search with confidence computation and indexing. Section 9.3 details the technique used for two- and multi-word queries. The phonetic search is covered in section 9.4 with a tri-phoneme approach to indexing described in section 9.5. Section 9.6 presents the experimental results in terms of index sizes and response-times evaluated on 17-hour subset of ICSI meeting database.

9.2 LVCSR-based search

LVCSR lattices (example in Fig. 17) contain nodes carrying word labels and arcs, determining the timing and acoustic (L_a^{lvcsr}) and language model (L_l^{lvcsr}) likelihoods generated by an LVCSR decoder. Usually, each speech record is first broken into segments (by speaker turn or voice activity detector) and each segment is represented by one lattice. The confidence of a keyword KW is given by

$$C^{lvcsr}(KW) = \frac{L_\alpha^{lvcsr}(KW) L^{lvcsr}(KW) L_\beta^{lvcsr}(KW)}{L_{best}^{lvcsr}}, \tag{5}$$

where the $L^{lvcsr}(KW) = L_a^{lvcsr}(KW)L_l^{lvcsr}(KW)$.

The forward likelihood $L_\alpha^{lvcsr}(KW)$ is the likelihood of the best path through lattice from the beginning of lattice to the keyword and the backward likelihood $L_\beta^{lvcsr}(KW)$ is the likelihood of the best path from the keyword to the end of lattice. For node N, these two likelihoods are computed by the standard Viterbi formulae:

$$L_\alpha^{lvcsr}(N) = L_a^{lvcsr}(N)L_l^{lvcsr}(N)\max_{N_P} L_\alpha^{lvcsr}(N_P) \tag{6}$$

$$L_\beta^{lvcsr}(N) = L_a^{lvcsr}(N)L_l^{lvcsr}(N)\max_{N_F} L_\beta^{lvcsr}(N_F) \tag{7}$$

where N_F is a set of nodes directly following node N (nodes N and N_F are connected by an arc) and N_P is a set of nodes directly preceding node N. The algorithm is initialized by setting $L_\alpha^{lvcsr}(first) = 1$ and $L_\beta^{lvcsr}(last) = 1$. The last likelihood we need in Eq. 5: $L_{best}^{lvcsr} = L_\alpha^{lvcsr} = L_\beta^{lvcsr}$ is the likelihood of the most probable path through the lattice.

The **indexing** of LVCSR lattices is inspired by [Brin and Page, 1998]. It begins with the creation of lexicon which provides a transformation from word to a unique number (ID) and vice versa. Then, a forward index is created storing each hypothesis (the word, its confidence, time and nodeID in the lattice file) in a hit list. From this index, a reverse index is created (like in text search) which has the same structure as the forward index, but is sorted by words and by confidence of hypotheses.

Each speech record (ie. meeting) is represented by many lattices. The reverse index tells us, in which lattice the keyword appears and what is it's nodeID in this particular lattice.

In the **search** phase, the reverse index is used to find occurrences of words from query. An important feature of our system is the generation of the most probable **context** of the found keyword – a piece of the Viterbi path from the found keyword forward and backward. For all matching occurrences, the searcher therefore loads into the memory a small part of lattice within which the found word occurs. Then, the searcher traverses this part of lattice in forward and backward directions selecting only the best hypotheses; in this way it creates the most probable string which traverses the found word.

9.3 Multi-word queries

A usable system for SDR should support queries of type

`word1 word2 word3` and `"word1 word2 word3"`

with the former one representing finding words in random order with optional spaces in between (in opposite to text-search where we work within a document, we specify a time-context) and the later one representing the exact match. Provided the query Q is found in the lattice, we again need to evaluate its confidence $C^{lvcsr}(Q)$. Similarly to Eq. 5, this is done by evaluating the likelihood of the path with all the words w_i belonging to the query and dividing it by the likelihood of the optimal path:

$$C^{lvcsr}(Q) = \frac{L_{rest}^{lvcsr}\prod_i L^{lvcsr}(w_i)}{L_{best}^{lvcsr}}, \tag{8}$$

where L_{rest}^{lvcsr} is the likelihood of the "Viterbi glue": optimal path from the beginning of the lattice to $w_{earliest}$, connections between words, w_i (for unquoted query) and optimal path from w_{latest} to the end of the lattice. In other words L_{rest}^{lvcsr} represents everything except the searched words. We should note, that each time we deviate the Viterbi path from the best one, we loose some likelihood, so that $L^{lvcsr}(Q)$ is upper-bounded by $\min_i C^{lvcsr}(w_i)$ — actually the confidence of the worst word in the query.

The same index as for single-word queries (keywords) is used here. Processing of a query involves the following steps:

1. Based on frequencies of words, the least frequent one from the query, w_{lf}, is taken as first and all its occurrences are retrieved.

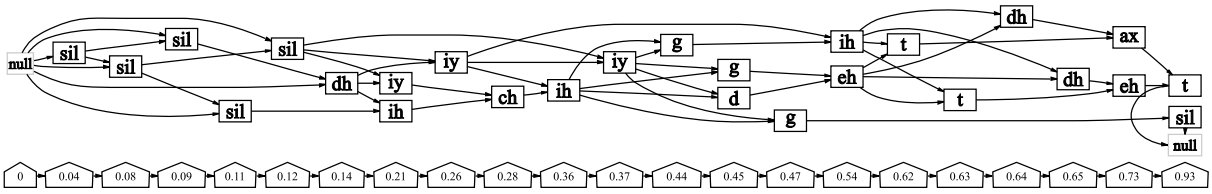

Figure 18: Example of a phoneme lattice

2. The search proceeds with other words and verifies if they are within the specified time interval from w_{lf} (for non-quoted queries) or joint to w_{lf} (for quoted ones). The internal memory representation resembles again a lattice. In such way, a candidate list is created.

3. The list is pre-sorted by the upper-bound of query confidence, as described above. The list is then limited to the pre-determined number of candidates (usually 10).

4. For these candidates, the evaluation of correct confidence is done according to Eq. 8. While looking for the "Viterbi glue", the Viterbi algorithm is extended before and after the part of lattice containing Q in order to obtain the left and right contexts.

9.4 Phonetic search

The main problem of LVSCR is the dependence on recognition vocabulary. The phonetic approach overcomes this problem by conversion of query to string and searching this string in a phoneme lattice (Fig. 18). The lattice has similar structure as word lattice (section 9.2), phonemes P populate nodes instead of words.

The confidence of keyword KW consisting of string of phonemes $P_b \dots P_e$ is defined similarly as in Eq. 5 by:

$$C^{phn}(KW) = \frac{L_\alpha^{phn}(P_b) L_\beta^{phn}(P_e) \prod_{P \in P_b \dots P_e} L_a(P)}{L_{best}^{phn}}, \tag{9}$$

where $L_\alpha^{phn}(P_b)$ is the forward Viterbi likelihood from the beginning of lattice to phoneme P_b, the product is the likelihood of the keyword, and $L_\beta^{phn}(P_e)$ is the likelihood from the last phoneme till the end of the lattice. L_{best} is the likelihood of the optimal path. As phoneme recognition is done without language (phono-tactic) model, the language model likelihoods are replaced by a constant – phoneme insertion penalty (PIP). It plays a role in the computation of $L_\alpha^{phn}(P_b)$, $L_\beta^{phn}(P_e)$ and L_{best}^{phn} and does not intervene in the product giving the likelihood of the keyword. The value of PIP needs to be tuned. The experiments of Szöke et al. [Szöke et al., 2005] have shown that in case the phoneme lattice is dense, it is sufficient to look for an exact match of the searched string and not to take into account substitution, insertion and deletion errors.

9.5 Indexing phoneme lattices

While the indexing of word lattices is straightforward, indexing phoneme lattices is more tricky: in advance, we do not know what we will search for. Yu and Seide in [Yu and Seide, 2005] and Siohan and Bacchiani in [Siohan and Bacchiani, 2005] have chosen indexing sequences of phonemes with variable length, we have however investigated a simpler approach making use of overlapping tri-phonemes and indexing similar to multi-word queries. The use of tri-phonemes was also recommended in [Ng, 2000] as the best balance between number of units and number of units' occurrences in a corpus.

In the indexing phase, tri-phonemes T_i are selected in lattices. For each T_i, its confidence is evaluated by Eq. 9 as if T_i was a keyword. In case this confidence is higher than a pre-determined threshold, the tri-phoneme is inserted into the index.

The search stage consists of the following steps:

1. The searched keyword generates a set of overlapping tri-phonemes. Based on their frequencies in the index, the least frequent one T_{lf}, is taken as first and all its occurrences are retrieved.

2. The search proceeds with other tri-phonemes and verifies that they form a chain in time (with a security margin between adjacent tri-phonemes). Similarly to multi-word queries, the internal memory representation has again the form of lattice. In such way, a candidate list is created.

3. The confidence of keyword is again upper-bounded by the confidence of the worst tri-phoneme. Based on these, the list is pre-sorted and limited to the pre-determined number of candidates (usually 10).

4. For these candidates, we go into the respective phoneme lattices and evaluate the correct confidence using Eq. 9.

We have verified, that in case no thresholds are applied in the index, we obtain exactly the same accuracy of search that in case phoneme lattices are processed directly.

9.6 Experiments

The evaluation was done on 17 hours of speech from ICSI meeting database [Janin et al., 2003]. Attention was paid to the definition of fair division of data into training, development and test parts with non-overlapping speakers. We have also balanced the ratio of native/nonnative speakers and balanced the ratio of European/Asiatic speakers.

LVCSR lattices were generated by AMI-LVCSR system [Hain et al., 2005a] and phonetic lattices were generated by a phoneme recognizer based on long-temporal context features with a hierarchical structure of neural nets [Schwarz et al., 2006].

The accuracies of different approaches were evaluated by Figure of Merit (FOM), which approximately corresponds to word accuracy provided that there are 5 false alarms per hour in average. In LVCSR-search, the FOM was 67% while for the phoneme-lattice search, we reached FOM of 60%. Detailed results are discussed in [Szöke et al., 2005] – in this experimental evaluation, we have concentrated on response times, and disk footprints that are crucial for real deployment of the system.

The size of audio is 1.8 GB. The number of LVCSR lattices representing this audio is 25815 and they occupy 600 MB. LVCSR index needs 130 MB. Phoneme lattices (branching factor 4) need 2.1 GB of disk and the tri-phoneme index requires 220 MB.

In all tests, we report average time to process one query. The number of hits was set to 10-best in all experiments. The context to retrieve in LVCSR queries was set to $\pm$10 words and $\pm$7.5 seconds (whichever is shorter). The processing was done on a AMD Athlon 3200+. We made sure that the data to be searched (lattices, indexes) resided on the local hard-disk and that no other CPU/memory consuming processes run on the machine.

The first test in LVCSR-search aimed at single keywords. Two sets were defined: `Test17` containing 17 frequent words and `Test1` containing words occurring just once in the test set. The total number of different words in `Test1` is 2310, but only 50 were used in these evaluations.

The following test aimed at 2- till 4-word *quoted* queries. We have randomly chosen sequences of 2 to 4 words from the transcriptions of the test set and made sure at least one word within each sequence is at least 5 characters long. Examples of such sequences are:
2: `"A MATTER", "NOUN PHRASES"`
3: `"THE DETECTOR TO", "PERSON TO DO"`
4: `"BUY A TICKET OR", "THE SITUATION OF LETTING"`
50 sequences of each length were selected. These tests are denoted `Quoted2 ... Quoted4`.

Test	time per query [s]
Test17	0.8
Test1	0.2
Quoted2	9.6
Quoted3	33.0
Quoted4	34.0
Unquoted2	1.2
Unquoted3	1.3
Unquoted4	1.8

Table 47: The results of LVCSR-based search.

Test	time per query [s]
Test17	10.5
Test1	9.3

Table 48: The results of phonetic search.

In the test of unquoted queries, all tested sequences contained only words with length ≥ 5 characters and we worked again with 2- till 4-word sequences (note that for unquoted sequence, the words can appear in any order). The context (or "document size") was set to 20 words. To define the sets of queries, we have divided the test set into windows containing 10 words, discarded windows with less than 10 words and selected one sequence satisfying the word-length constraint from each window. Then, these sequences were randomized and 50 were selected for each length. Examples of such sequences are:
2: RELEVANT RANGES, WEDNESDAY ACTUAL
3: PERSON LISTENING FIRST, STUDY RIGHT GERMANY
4: TEACHER QUALITY THERE COURSE, TRAIN MODELS SUBTRACTION USING
These tests are denoted Unquoted2 ... Unquoted4.

Table 47 summarizes the response times for LVCSR-based search.

In the tests of phonetic search, only single keywords from sets Test17 and Test1 were looked for. Measurement of response times were done on the same 17h test-set, the results are summarized in Table 48.

We see that in LVCSR search is very fast and that single word and multiple-word unquoted queries require only 1-2 seconds. It is very likely that these figures will extend well to bigger archives. The times required for quoted queries are quite prohibitive and we need to suggest optimizations. One of first targets will be the C++ STL library that is used for the creation of the internal lattice structures and which is quite slow.

The response times of phonetic search are longer than in LVCSR, but the search is still usable for the given size of archive. We should note that the comparison of response times for Test17 and Test1 is inverse for LVCSR and phonetic search. This is explained by the nature of the two algorithms: LVCSR can take advantage of rarity of words in Test1 – they simply appear less frequently in the index so that the processing is faster. On contrary, phonetic search of items from Test1 takes almost the same time as Test17, as this approach can do no difference between rare and frequent words (actually, it does not have a notion of "word" in both indexing and search).

9.7 Conclusions

We have presented several techniques of indexing and search in LVCSR and phonetic lattices for spoken document retrieval. They were evaluated on real meeting data from ICSI meeting database. In LVCSR, both one-word and multi-word queries are handled with fast response times, the processing of quoted queries still needs some

investigation. In phonetic search, we have verified the functionality of indexing tri-phones derived from phoneme lattices, but speeding up is needed also here.

In our further research, we will investigate direct techniques to derive tri-phoneme indices without lattices - tri-phonemes can actually be seen as keywords and as such pre-detected by a standard acoustic keyword spotting and indexed. We will also investigate the importance of different tri-phonemes for indexing and search and suggest customized pruning thresholds to keep the index size manageable. Finally, our goal is to build and test a system combining LVCSR and phonetic search allowing to search multi-word queries with OOVs.

10 AMI Hotspots Status and Planning: Visual Features

10.1 Introduction

The focus of our activities has been to extract visual features which can be used (in combination with audio features) for 'hotspot' segementation i.e. the detection of events such as 'opening a meeting', 'laughter', etc. For feature extraction we have split the visual information available in the video stream into 3 classes, texture (appearance), colour and motion, and investigated these seperately. Our next step will be to integrate these features. For tracking we have investigated 3D reconstruction.Our work sofar has culminated in a feature browser (section 10.8).

10.2 Textural Features

Faces are rich in hotspot-related information and constitute one of the few classes of textural feature which is common to all meetings. We have experimented with different face detectors [36] [37] and used the probability of a face detection in the video as a feature, The face detection algorithms used are designed to detect frontal views of a face and will not detect a face if the gaze direction of the subject is too far from the optical axis of the video camera. Therefore, this feature has some dependence on gaze direction as well as the presence or absence of a person. In a trial experiment involving a single meeting it was noted that a simultaneous transition from face to no-face in multiple individual cameras co-incided with each member of the group turning to look at the projected slides, thus indicating a change in the focus of attention of the group. Note that this feature is dependent on the meeting room geometry. The degree to which subject must turn to see a projected slide depends on the seating arrangement and in order to address another participant a subject usually turns his/her head and this can affect the feature.Due of their maturity and robustness, face detectors are also commonly used as a starting point for segmenting and analysing sub-regions of a face. Eye position, for example, is also estimated as standard with many tools. We have begun to investigate the use of face detectors to segment mouths with the intention of extracting features which can be used to indicate the emotional state of the subject, for example, with the aim of combining visual features with audio laughter detectors to make them more robust. Face detectors can also provide constraints for skin-based colour models as discussed below.

10.3 Colour Features

Skin hues are common to all meetings and skin-colour models can be used to segment faces hands and (sometimes) arms. We have implemented a colour model that segments skin by fitting a single, two-dimensional Gaussian to a set of training samples in YCrCb colour space [Gong and Sakauchi, 1995]. We have optimised our implementation with a look-up table to allow skin to be segmented efficiently. Obviously, background objects which are the same colour as skin can cause problems for colour-only segmentation and unfortunately this is the case for some AMI recordings where the background wall colour is very similar to skin. One way we can improve our skin-segmentation algorithm is by learning meeting-specific skin models adaptively by using face detectors to automatically add skin samples to the colour model.We have divided the video frame into subregions and used the fraction of skin pixels in each region as a feature. Trial experiments showed showed that some actions, such as hand-to-head motions, could be identified using these features.In the future we intend to use skin-segmentation to improve tracking performance for heads and hands.

10.4 Motion Features

Motion is a useful cue for synchronised actions and certain types of hotspots. We have used background subtraction techniques to measure the amount of activity in different subregions of the video frame and used the fraction of

[36] Machine Perception Toolbox version 1.0 `http://mplab.ucsd.edu/grants/project1/freesoftware/mptwebsite/API/index.html`
[37] FaceIT SDK `http://www.identix.com`

moving pixels as a feature. In trial experiments it was found that periods in which motion features from multiple participants were synchronised often indicated meeting hotspots, such as laughter. Laughter is accompanied by lots of body motion and has a strong social function, when one person laughs most participants join in and this can produce clear signals in our combined low-level motion features.Body motion is a good indicator that a position at the meeting table is occupied (better than face detection that depends on gaze direction) and is often correlated with active participation in a meeting. When people talk they, literally, become animated and produce strong motion signals. We are preparing a large set of motion features that will be comibined with audio features to improve laughter detection and speaker identification.

10.5 Tracking

We have used overview cameras with overlapping fields of view to investigate 3D tracking techniques. In particular, by using algorithms which use epipolar constraints to estimate the fundamental matrix that allows pixels positions in two camera images to be projected into 3D co-ordinates. The main challenge for these techniques is not establishing the fundamental matrix, which can be reliably estimated with limited user input, but dealing with the changes in appearance of an object that is caused by the different viewpoints of the cameras. Our focus is on finding features that are robust to changes in viewpoint and allow proper 3D tracking.

10.6 Visualisation/Data Exchange

One challenge faced for multimodal algorithm development with video data is the appropriate visualisation of the algorithm output. Another involves the efficient exchange of data between reserachers. Part of our activities has been focussed on extending the functionality of a visualisation tool, the 'segment viewer', to allow multiple features to be viewed and the associated video fragments to be played easily by different researchers using the same protocol.

10.7 Future Work

Our next step will be to integrate features we have been experimenting with so far. For example by using a face detector to improve skin segmentation and using skin detection to improve motion analysis and tracking. We will generate visual features for larger datasets so that they can be combined with audio features and the effect of combining visual features with audio features can be quantified. Of particular interest is our observation that the combined behaviour of multiple low-level features can be used to identify interesting group activities and we will investigate methods to combine these low-level features for classification.

10.8 Feature Browser

A feature browser has been implemented in Matlab that allows for manual inspection of alleged hot spots in ameeting. In this browser, on a horizontal time scale, the amount of motion extracted from video for each person and for each motion zone [Gong and Sakauchi, 1995]is plotted. Aligned motion activity indicates possible hot spots; motion ofa dedicated person such as the chairman may also provide important cues for e.g, topic segmentation.

Figure 19: Motion zones per speaker

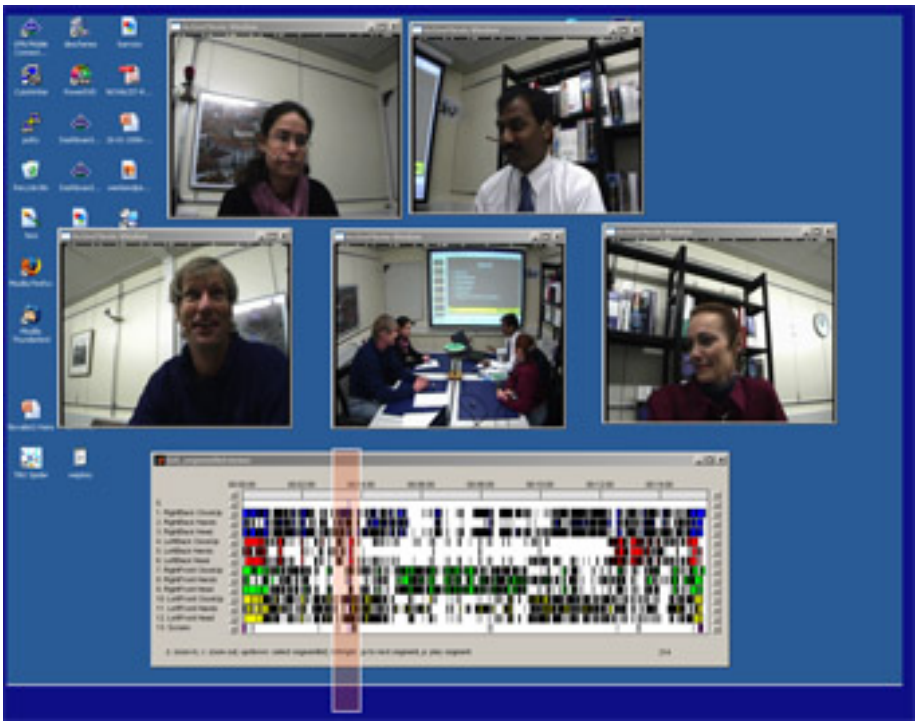

Figure 20: Feature browser closeup view.

11 Future Work

This version of the deliverable presents fully implemented systems, with clearly defined tasks and evaluation procedures as well as initial results, mostly on hand-annotated input. Most systems are also working with AMI hub scenario data now, after initial versions were mostly using the ICSI corpus as it was available from the very beginning of the AMI project.

We will publish a revised version of this deliverable that will include the results of the experiments with actual ASR output and other, fully automatically generated features and their discussion. This version is planned for the end of November 2006.

In addition to the automatic system, some tasks will report two other types of results:

(1) results for the same kinds of features, but that use reference/gold standard versions of the features (reference transcription, hand-coded dialogue acts, etc.). This will allow us to judge the degradation we have to expect when applying our systems in an on-line setting, running directly from automatically generated transcripts and other features. Initial experiments, e.g. in Dialogue Act recognition, indicate that degradation of the quality of our systems roughly corresponds to the degradation in transcription quality. Note that using multiple (and multi-modal) input sources beyond the transcript makes the systems more robust and thus implicitly can actually correct ASR errors;

(2) results making use of non-automatic features for which we have no automatic processes (or where it would be too expensive to do the automatic version), showing whether or not adding the feature to what we can get automatically provides enough of an improvement that it would be worth attempting automatic processing in future.

In the final months of AMI, we will work towards fully automated systems as outlined above. Most systems will be improved further, running extended experiments with various ML algorithms and feature sets. Work in abstractive summarization is ongoing, with more of the necessary annotations (e.g. ontology-based propositional content) and automated systems (e.g. semantic parsing) becoming available.

We will also finish currently ongoing work in a number of areas, in particular

- influence classification
- keyword spotting and indexing (see also the work already reported in D5.1)
- classification of meeting activities using conversational state sequences
- decision point detection
- sentence compression in extractive summarization

Decision Point Detection We are working on the task of detecting decision-making points from a long sequence of utterances. Current work makes no distinction of training and test set and evaluation is performed as leave-one-out cross validation.

The system will be trained on specialized abstractive summaries containing decisions, problems, and action items (as sentences) and linked extractive summaries (from dialogue acts to decision sentences). We use a feature-based approach (MaxEnt) to train a classification model to assign each end of utterance into the class of decision-making point or non-decision making point.

A large list of features is used:

- Lexical features: language models (word, bigram, trigram), selected cue phrases based on association with decision-making points, decision-making orientation; unit length, # of subjective words (Wilson), POS tags of the first and the last phrase

- Prosodic features: duration, speech rate (# of words spoken per second), silence, energy (average of every quarter), pitch (maximum F0, contour, variance), pitch slope at multiple points within a dialogue act, prosodic prominence
- Dialogue act type
- Adjacency pair features: Type, # of overlapping words between an adjacency pair, # of conversation units in-between an AP
- AMI meta-data features: Speaker type, meeting type
- Contextual (Proximity) features: the speaker of the preceding unit and the following unit, preceding and following dialogue act type, target and source dialogue act in it associated adjacency pair, position in a meeting, position to the last most likely decision points
- Topic related features: Topic label, occurring in subtopic segments or not, position from topic shifts, topical novelty
- Argumentation structure-related features: Position in a discussion
- Focus of attention and Individual actions
- Other more abstract-level pragmatic phenomenon (e.g., dominance level, group-level interest, hot spot, agreement/disagreement)

Evaluation will be an N-fold leave-one-out cross validation, i.e., the system will be trained on N-1 of the N meetings and tested on the remaining one meeting; this will be repeated N times. The main evaluation metrics will be precision, i.e., the accuracy of decision-making point detection, and recall, i.e., the coverage of decision summary sentences. A first reliability test with extractive summaries on three sets of meetings (ES2008, IS1003, TS3005) and two coders resulted in an average kappa value of 0.43.

References

[AMI, 2005] (2005). Guidelines for dialogue act and addressee annotation version 1.0. http://www.idiap.ch/amicorpus/documentations.

[Abney, 1991] Abney, S. (1991). Parsing by chunks. In Berwick, R. C., Abney, S. P., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer Academic Publishers, Boston.

[Abney, 1996] Abney, S. (1996). Partial parsing via finite-state cascade. *Natural Language Engineering*, 2(4):337–344.

[Andernach, 1996] Andernach, T. (1996). A machine learning approach to the classification of dialogue utterances. *Computing Research Repository*, july.

[Ando and Zhang, 2005] Ando, R. and Zhang, T. (2005). A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 1–9, Ann Arbor, Michigan. Association for Computational Linguistics.

[Ang et al., 2005] Ang, J., Liu, Y., and Shriberg, E. (2005). Automatic dialog act segmentation and classification in multiparty meetings. In *Proc. ICASSP*, volume 1, pages 1061–1064, Philadelphia, USA.

[Austin, 1962] Austin, J. L. (1962). *How to do Things with Words*. Oxford: Clarendon Press.

[Banerjee and Pedersen, 2003] Banerjee, S. and Pedersen, T. (2003). The design, implementation, and use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City.

[Barzilay and Lapata, 2005] Barzilay, R. and Lapata, M. (June 2005). Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, MI, USA*.

[Beeferman et al., 1999] Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. *Machine Learning*, 34:177–210.

[Berger, 1997] Berger, A. (1997). Convexity, maximum likelihood and all that. http://www.cs.cmu.edu/ aberger/maxent.html.

[Berger et al., 1996] Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[Bilmes, 2000] Bilmes, J. (2000). Dynamic bayesian multinets. *Proc. Int. Conf. on Uncertainty in Artificial Intelligence*.

[Bilmes and Kirchhoff, 2003] Bilmes, J. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. *Proceedings of HLT/NAACL 2003*.

[Bilmes and Zweig, 2002] Bilmes, J. and Zweig, G. (2002). The Graphical Model ToolKit: an open source software system for speech and time-series processing. *Proc. IEEE ICASSP*.

[Blei and Moreno, 2001] Blei, D. M. and Moreno, P. J. (2001). 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.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Technical report, Computer Science Department, Stanford University.

[Carbonell and Goldstein, 1998] Carbonell, J. and Goldstein, J. (August 1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia*, pages 335–336.

[Carletta et al., 2006] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., , and Wellner, P. (2006). The AMI meeting corpus: A pre-announcement. In *Proceedings of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*. Springer.

[Carletta et al., 2005] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., and Wellner, P. (2005). The ami meetings corpus. In *Proceedings of the Measuring Behavior 2005 symposium on "Annotating and measuring Meeting Behavior"*. AMI-108.

[Carreras et al., 2005] Carreras, X., Màrquez, L., and Castro, J. (2005). Filtering-ranking perceptron learning for partial parsing. *Machine Learning*, 60:41–71.

[Cathcart et al., 2003] Cathcart, N., Carletta, J., and Klein, E. (2003). A shallow model of backchannel continuers in spoken dialogue. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 51–58, Morristown, NJ, USA. Association for Computational Linguistics.

[Cheng and Greiner, 1999] Cheng, J. and Greiner, R. (1999). Comparing bayesian network classifiers. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 101–108, San Francisco, CA. Morgan Kaufmann Publishers.

[Choi, 2000] Choi, F. (2000). Advances in domain independent linear text segmentation. *Proceedings of NAACL*, pages 26–33.

[Choi et al., 2001] Choi, F., Wiemer-Hastings, P., and Moore, J. D. (2001). Latent semantic analysis for text segmentation. In Lee, L. and Harman, D., editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 109–117.

[Christensen et al., 2005] Christensen, H., Kolluru, B., Gotoh, Y., and Renals, S. (2005). Maximum entropy segmentation of broadcast news. In *Proceedings of the IEEE International Conference on Acoustic Speech and Signal Processing*, Philadelphia USA.

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

[Cooper and Herskovits, 1992] Cooper, G. and Herskovits, E. (1992). Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

[Cover and Thomas, 1991] Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA.

[Darroch and Ratcliff, 1972] Darroch, J. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43:1470–1480.

[Daumé III and Marcu, 2006] Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

[Dhillon et al., 2004] Dhillon, R., Bhagat, S., Carvey, H., and Shriberg, E. (2004). Meeting recorder project: Dialogue act labeling guide. Technical Report TR-04-002, International Computer Science Institute (ICSI), Berkeley, CA.

[Dielmann and Renals, 2006] Dielmann, A. and Renals, S. (2006). Multistream recognition of dialogue acts in meetings. In *Machine Learning for Multimodal Interaction: 3rd International Workshop, MLMI 2006*, Washington, USA.

[E. Hovy and Fukumoto, 2006] E. Hovy, C.Y. Lin, L. Z. and Fukumoto, J. (2006). Automated summarization evaluation with basic elements. In *Proc. of LREC 2006, Genoa, Italy*.

[Fernandez and Picard, 2002] Fernandez, R. and Picard, R. (2002). Dialog act classification from prosodic features using support vector machines. In *Proceedings of speech prosody 2002*.

[Foltz et al., 1998] Foltz, P., Kintsch, W., and Landauer, T. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25.

[Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, (29):131–163.

[Galley et al., 2003] Galley, M., McKeown, K., Fosler-Lussier, E., and Jing, H. (2003). Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

[Gildea, 2001] Gildea, D. (2001). Corpus variation and parser performance. In Lee, L. and Harman, D., editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202.

[Godfrey et al., 1992] Godfrey, J., Holliman, E., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, San Francisco.

[Gong and Liu, 2001] Gong, Y. and Liu, X. (September 2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana, USA*, pages 19–25.

[Gong and Sakauchi, 1995] Gong, Y. and Sakauchi, M. (1995). Detection of regions matching specified chromatic features. *Computer Vision and Image Understanding*, 61(2):263–269.

[Gotoh and Renals, 2000a] Gotoh, Y. and Renals, S. (2000a). Information extraction from broadcast news. *Philosophical Transactions of the Royal Society of London, Series A*, 358:1295–1310.

[Gotoh and Renals, 2000b] Gotoh, Y. and Renals, S. (2000b). Sentence boundary detection in broadcast speech transcripts. In *Proc. of ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000*, pages 228–235.

[Grice, 1969] Grice, H. P. (1969). Utterer's meaning and intentions. *Philosophical Review*.

[Grosz and Hirschberg, 1992] Grosz, B. and Hirschberg, J. (1992). Some intonational characteristics of discourse structure. In *Proceedings of the International Conference on Spoken Language Processing*.

[Hachey et al., 2005] Hachey, B., Murray, G., and Reitter, D. (October 2005). The Embra system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Proceedings of the Document Understanding Conference (DUC) 2005, Vancouver, BC, Canada*.

[Hain et al., 2005a] Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., McCowan, I., Moore, D., Wan, V., Ordelman, R., and Renals, S. (2005a). The 2005 AMI system for the transcription of speech in meetings. In *Proc. Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh.

[Hain et al., 2005b] Hain, T., Karafit, M., Garau, G., Moore, D., Wan, V., Ordelman, R., and Renals, S. (2005b). Transcription of conference room meetings: an investigation. *Proc. Interspeech 2005 - Eurospeech, Lisbon*.

[Halliday and Hasan, 1976] Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. London: Longman.

[Hastie et al., 2002] Hastie, H., Poesio, M., and Isard, S. (2002). Automatically predicting dialogue structure using prosodic features. *Speech Communication*, (36):63–79.

[Hearst, 1997] Hearst, M. (1997). TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics 23(1)*, pages 33–64.

[Hirschberg and Litman, 1993] Hirschberg, J. and Litman, D. (1993). Empirical studies on the disambiguation of cue phrases. *Comput. Linguist.*, 19(3):501–530.

[Hsueh et al., 2006] Hsueh, P.-Y., Moore, J., and Renals, S. (2006). Automatic segmentation of multiparty dialogue. In *the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

[Huang and Zweig, 2002] Huang, J. and Zweig, G. (2002). Maximum entropy model for punctuation annotation from speech. In *Proc. ICSLP*, pages 917–920, Denver, USA.

[J. Yamron et al., 1998] J. Yamron, I., Carp, I., Gillick, L., Lowe, S., and van Mulbregt, P. (1998). A hidden markov model approach to text segmentation and event tracking. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 333–336.

[Janin et al., 2003] Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., and Wooters, C. (2003). The ICSI meeting corpus. In *Proceedings of IEEE ICASSP 2003, Hong Kong, China*, pages 364–367.

[Jelinek, 1990] Jelinek, F. (1990). Self-organized language modeling for speech recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann Publishers, Inc.

[Ji and Bilmes, 2005] Ji, G. and Bilmes, J. (2005). Dialog act tagging using graphical models. In *Proc. ICASSP*, volume 1, pages 33–36, Philadelphia, USA.

[Jovanovic et al., 2005] Jovanovic, N., op den Akker, R., and Nijholt, A. (2005). A corpus for studying addressing behavior in multi-party dialogues. In *Proceeding of the 6th SIGDial Workshop on Discourse and Dialogue*.

[Jurafsky et al., 1997] Jurafsky, D., Shriberg, E., and Biaska, D. (1997). Switchboard SWBD-DAMSL shallow-discourse-function annotation (coders manual, draft 13). Technical report, Univ. of Colorado, Inst. of Cognitive Science.

[Jurafsky et al., 1998] Jurafsky, D., Shriberg, E., Fox, B., and Curl, T. (1998). Lexical, prosodic, and syntactic cues for dialog acts. In Stede, M., Wanner, L., and Hovy, E., editors, *Discourse Relations and Discourse Markers: Proceedings of the Conference*, pages 114–120. Association for Computational Linguistics, Somerset, New Jersey.

[Kanselaar et al., 2003] Kanselaar, G., Erkens, G., Andriessen, J., Prangsma, M., Veerman, A., and Jaspers, J. (2003). *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, chapter Desiging Argumentation Tools for Collaborative Learning. Springer Verlag, London, UK.

[Katrenko, 2004] Katrenko, S. (2004). Textual data categorization: back to the phrase-based representation. In *Proceedings in 2nd International IEEE Conference "Intelligent systems", Vol. III*, pages 64–67.

[Keizer and Akker, 2005] Keizer, S. and Akker, R. o. d. (2005). Dialogue act recognition under uncertainty using bayesian networks. *Natural Language Engineering*, 1:1–30.

[Kirchhoff et al., 2002] Kirchhoff, K., Bilmes, J., Henderson, J., Schwartz, R., Noamany, M., Schone, P., Ji, G., Das, S., Egan, M., He, F., Vergyri, D., Liu, D., and Duta, N. (2002). Novel approaches to arabic speech recognition - final report from the jhu summer workshop 2002. *Tech. Rep., John-Hopkins University*.

[Klein and Manning, 2003] Klein, D. and Manning, C. (2003). Maxent models, conditional estimation, and optimization.
http://nlp.stanford.edu/downloads/classifier.shtml. HLT,-NAACL 2003 and ACL 2003 Tutorial.

[Kudo, 2006] Kudo, T. (2006). `http://chasen.org/ taku/software/CRF++/`.

[Kudo and Matsumoto, 2000] Kudo, T. and Matsumoto, Y. (2000). Use of support vector learning for chunk identification. In Cardie, C., Daelemans, W., Nedellec, C., and Tjong Kim Sang, E., editors, *Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144. Lisbon, Portugal.

[Kudo and Matsumoto, 2001] Kudo, T. and Matsumoto, Y. (2001). Chunking with support vector machines. In *Proceedings of NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Kupiec et al., 1995] Kupiec, J., Pederson, J., and Chen, F. (July 1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA*, pages 68–73.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

[Landauer and Dumais, 1997] Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

[Lendvai et al., 2003] Lendvai, P., Bosch, A. v. d., and Krahmer, E. (2003). 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.

[Lin and Hovy, 2003] Lin, C.-Y. and Hovy, E. H. (May 2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL 2003, Edmonton, Calgary, Canada*.

[Litman and Passoneau, 1995] Litman, D. and Passoneau, R. (1995). Combining multiple knowledge sources for discourse segmentation. In *Proceedings of the ACL*.

[Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large-scale optimization. *Math. Programming*, 45:503–528.

[Liu et al., 2005] Liu, Y., Shriberg, E., Stolcke, A., Peskin, B., Ang, J., Hillard, D., Ostendorf, M., Tomalin, M., Woodland, P., and Harper, M. (2005). Structural metadata research in the EARS program. In *Proc. of ICASSP*, volume 5, pages 957–960.

[Makhoul et al., 1999] Makhoul, J., Kubala, F., Schwartz, R., and Weischedel, R. (1999). Performance measures for information extraction. In *Proceedings of DARPA Broadcast News Workshop*, pages 249–252.

[Malouf, 2002] Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. In *Proc. Conference on Natural Language Learning, CoNLL*, pages 49–55.

[Mann and Thompson, 1988] Mann, W. and Thompson, S. (1988). Rhetorical structure theory: toward a functional theory of text organization. *Text*, 8:243–281.

[Marcus et al., 1993] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

[Maskey and Hirschberg, 2005] Maskey, S. and Hirschberg, J. (September 2005). Comparing lexial, acoustic/prosodic, discourse and structural features for speech summarization. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Lisbon, Portugal*.

[Mast et al., 1996] Mast, M., Kompe, S., Harbeck, A., Kiessling, H., Niemann, E., and Nöth, E. (1996). Dialog act classification with the help of prosody. In *Proc. ICSLP*, volume 3, pages 1732–1735, Philadelphia, USA.

[Morris and Hirst, 1991] Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1).

[Murphy, 2002] Murphy, K. P. (2002). Dynamic Bayesian networks: Representation, inference and learning. *Ph.D. Thesis, UC Berkeley, Computer Science Division*.

[Murray et al., 2005a] Murray, G., Renals, S., and Carletta, J. (September 2005a). Extractive summarization of meeting recordings. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Lisbon, Portugal*.

[Murray et al., 2005b] Murray, G., Renals, S., Carletta, J., and Moore, J. (June 2005b). Evaluating automatic summaries of meeting recordings. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, MI, USA*.

[Nagata and Morimoto, 1993] Nagata, M. and Morimoto, T. (1993). 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.

[Nagata and Morimoto, 1994] Nagata, M. and Morimoto, T. (1994). First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15:193–203.

[Nenkova and Passonneau, 2004] Nenkova, A. and Passonneau, B. (May 2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of HLT-NAACL 2004, Boston, MA, USA*.

[Ng, 2000] Ng, K. (2000). *Subword-Based Approaches for Spoken Document Retrieval*. PhD thesis, Massachusetts Institute of Technology.

[NIST website, 2003] NIST website (2003). Rt-03 fall rich transcription. http://www.nist.gov/speech/tests/rt/rt2003/fall/.

[Osborne, 2000] Osborne, M. (2000). Shallow parsing as part-of-speech tagging. In Cardie, C., Daelemans, W., Nedellec, C., and Tjong Kim Sang, E., editors, *Proceedings of CoNLL-2000 and LLL-2000*, pages 145–147, Lisbon, Portugal.

[Osborne, 2002] Osborne, M. (2002). Shallow parsing using noisy and non-stationary training material. *Journal of Machine Learning Research*, 2:695–719.

[Pallotta et al., 2005] Pallotta, V., Niekrasz, J., and Purver, M. (2005). Collaborative and argumentative models of meeting discussions. In *Proceeding of CMNA-05 international workshop on Computational Models of Natural Arguments (part of IJCAI 2005)*.

[Pevzner and Hearst, 2002] Pevzner, L. and Hearst, M. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, pages 19–36.

[Ponte and Croft, 1997] Ponte, J. M. and Croft, W. B. (1997). Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*.

[Quinlan, 1993] Quinlan, J. (1993). *C4.5 : programs for machine learning*. Morgan Kaufmann, San Mateo, CA, USA.

[Radev et al., 2001] Radev, D., Blair-Goldensohn, S., and Zhang, Z. (September 2001). Experiments in single and multi-document summarization using mead. In *The Proceedings of the First Document Understanding Conference, New Orleans, LA*.

[Ramshaw and Marcus, 1995] Ramshaw, L. and Marcus, M. (1995). Text chunking using transformation-based learning. In Yarovsky, D. and Church, K., editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94.

[Ratnaparkhi, 1996] Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In Brill, E. and Church, K., editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing 1996*, pages 133–142.

[Ratnaparkhi, 1998] Ratnaparkhi, A. (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA.

[Reidsma et al., 2005] Reidsma, D., Hofs, D., and Jovanovic, N. (2005). A presentation of a set of new annotation tools based on the nxt api. Poster at Measuring Behaviour 2005. AMI-105.

[Reithinger and Klesen, 1997] Reithinger, N. and Klesen, M. (1997). Dialogue act classification using language models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, Rhodes, Greece.

[Reynar, 1998] Reynar, J. (1998). *Topic Segmentation: Algorithms and Applications*. PhD thesis, UPenn, PA USA.

[Rienks and Heylen, 2005] Rienks, R. and Heylen, D. (2005). Argument diagramming of meeting conversations. In Vinciarelli, A. and Odobez, J.-M., editors, *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces (ICMI)*, pages 85–92, Trento, Italy.

[Rienks et al., 2006] Rienks, R., Nijholt, A., and Barthelmess, P. (2006). Pro-active meeting assistants : Attention please! In *Social Intelligence Design*, Osaka, Japan.

[Ries, 1999] Ries, K. (1999). HMM and neural network based speech act detection. In *Proc. ICASSP*, volume 1, pages 497–500, Phoenix, USA.

[Roark et al., 2006] Roark, B., Liu, Y., Harper, M., Stewart, R., Lease, M., Snover, M., Shafran, I., Dorr, B., Hale, J., Krasnyanskaya, A., and Yung, L. (2006). Reranking for sentence boundary detection in conversational speech. In *Proc. of ICASSP*, volume 1, pages 545–548.

[Rosset and Lamel, 2004] Rosset, S. and Lamel, L. (2004). Automatic detection of dialog acts based on multi-level information. In *Procedings of the ICSLP*, pages 540–543, Jeju Island, Korea.

[Rotaru, 2002] Rotaru, M. (2002). Dialog act tagging using memory-based learning. Technical report, University of Pittsburgh. Term project in Dialogue-Systems class.

[Samuel, 2000] Samuel, K. (2000). *Discourse Learning: An Investigation of Dialogue Act Tagging using Transformation-Based Learning.* PhD thesis, Department of Computer and Information Sciences. University of Delaware. Newark, Delaware.

[Samuel et al., 1998] Samuel, K., Carberry, S., and Vijay-Shanker, K. (1998). Dialogue act tagging with transformation-based learning. In *Proc. 17th Int. Conference on Computational Linguisitics*, volume 2, pages 1150–1156, Montreal, Canada.

[Samuel et al., 1999] Samuel, K., Carberry, S., and Vijay-Shanker, K. (1999). Automatically selecting useful phrases for dialogue act tagging. *The Computing Research Repository*.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168.

[Schum and Martin, 1982] Schum, D. and Martin, A. (1982). Formal and empirical research on cascaded inference in jurisprudence. *Law and Society Review*, 17(1):105–152.

[Schwartz et al., 2001] Schwartz, R., Sista, S., and Leek, T. (2001). Unsupervised topic discovery. In *Proceedings of Workshop on Language Modeling and Information Retrieval*.

[Schwarz et al., 2006] Schwarz, P., Matějka, P., and Černocký, J. (2006). Hierarchical structures of neural networks for phoneme recognition. In *Proc. ICASSP 2006*, Toulouse, France.

[Searle, 1969] Searle, J. (1969). *Speech Acts*. Cambridge University Press.

[Sha and Pereira, 2003] Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141, Morristown, NJ, USA. Association for Computational Linguistics.

[Shriberg et al., 1998] Shriberg, E., Bates, R., Taylor, P., Stolcke, A., Jurafsky, D., Ries, K., Coccaro, N., Martin, R., Meteer, M., and Ess-Dykema, C. V. (1998). Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, (41):439–487.

[Shriberg et al., 2004] Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., , and Carvey, H. (April-May 2004). The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue, Cambridge, MA, USA*, pages 97–100, Cambridge, USA.

[Shriberg et al., 2000] Shriberg, E., Stolcke, A., Hakkani-Tür, D., and Tür, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154.

[Shum, 1997] Shum, S. (1997). Negotiating the construction and reconstruction of organisational memories. *Journal of Universal Computer Science*, 3(8):899–928.

[Siohan and Bacchiani, 2005] Siohan, O. and Bacchiani, M. (2005). Fast vocabulary-independent audio search using path-based graph indexing. In *Proc. Eurospeech 2005*, Lisbon, Portugal.

[Sparck-Jones, 1998] Sparck-Jones, K. (1998). Automatic summarising: factors and directions. In Mani, I. and Maybury, M., editors, *Advances in Automatic Text Summarization*. MIT press.

[Steedman, 2000] Steedman, M. (2000). *The syntactic process*. MIT Press, Cambridge, MA, USA.

[Steidl et al., 2005] Steidl, S., Levit, M., Batliner, A., Nöth, E., and Niemann, H. (2005). "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*.

[Steinberger and Ježek, 2004] Steinberger, J. and Ježek, K. (April 2004). Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of ISIM 2004, Roznov pod Radhostem, Czech Republic*, pages 93–100.

[Stokes et al., 2004] Stokes, N., Carthy, J., and Smeaton, A. (2004). Select: a lexical cohesion based news story segmentation system. *AI Communications*, 17(1):3–12.

[Stolcke, 2002] Stolcke, A. (2002). SRILM an extensible language modeling toolkit. *Proc. Int. Conf. on Spoken Language Processing*.

[Stolcke et al., 2006] Stolcke, A., Anguera, X., Boakye, K., Çetin, O., Grezl, F., Janin, A., Mandal, A., Peskin, B., Wooters, C., and Zheng, J. (2006). Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system. In Renals, S. and Bengio, S., editors, *Machine Learning for Multimodal Interaction: 2nd International Workshop, MLMI 2005*, pages 463–475. LNCS 3869, Springer.

[Stolcke et al., 2000] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26:339–373.

[Stolcke and Shriberg, 1996] Stolcke, A. and Shriberg, E. (1996). Automatic linguistic segmentation of conversational speech. In *Proc. ICSLP*, volume 2, pages 1005–1008, Philadelphia, USA.

[Szöke et al., 2005] Szöke, I., Schwarz, P., Matějka, P., Burget, L., Karafiát, M., Fapšo, M., and Černocký, J. (2005). Comparison of keyword spotting approaches for informal continuous speech. In *Proc. Eurospeech 2005*, Lisbon, Portugal.

[Tjong Kim Sang and Buchholz, 2000] Tjong Kim Sang, E. F. and Buchholz, S. (2000). Introduction to the conll-2000 shared task: Chunking. In Cardie, C., Daelemans, W., Nedellec, C., and Tjong Kim Sang, E., editors, *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal.

[Tjong Kim Sang and De Meulder, 2003] Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

[Toutanova et al., 2003] Toutanova, K., Klein, D., and Manning, C. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Morristown, NJ, USA. Association for Computational Linguistics.

[U.M.Fayyad and Irani, 1995] U.M.Fayyad and Irani, K. (1995). Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on AI*, pages 194–202.

[Utiyama and Isahara, 2001] Utiyama, M. and Isahara, H. (2001). A statistical model for domain-independent text segmentation. In *Proceedings of the 28th Annual Meeting of the ACL*.

[Valenza et al., 1999] Valenza, R., Robinson, T., Hickey, M., and Tucker, R. (April 1999). Summarization of spoken audio through information extraction. In *Proceedings of the ESCA Workshop on Accessing Information in Spoken Audio, Cambridge UK*, pages 111–116.

[van den Bosch, 2004] van den Bosch, A. (2004). Wrapped progressive sampling search for optimizing learning algorithm parameters. In *R. Verbrugge, N. Taatgen, and L. Schomaker (Eds.) Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*.

[van den Bosch and Daelemans, 2005] van den Bosch, A. and Daelemans, W. (2005). Improving sequence segmentation learning by predicting trigrams. *Proceedings of the Ninth Conference on Natural Language Learning, CoNLL-2005*, pages 80–87.

[Van Gelder, 2001] Van Gelder, T. (2001). How to improve critical thinking using educational technology. In *Proceedings of the 18th annual conference of the Australasian Society for Computers in Learning in Tertiary education*, pages 539–548.

[van Mulbregt et al., 1999] van Mulbregt, P., Carp, J., Gillick, L., Lowe, S., and Yamron, J. (1999). Segmentation of automatically transcribed broadcast news text. In *Proceedings of the DARPA Broadcast News Workshop*, pages 77–80. Morgan Kaufman Publishers.

[van Rijsbergen, 1979] van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworths, 2 edition.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley and Sons.

[Veerman, 2000] Veerman, A. (2000). *Computer-supported collaborative learning through argumentation*. PhD thesis, University of Utrecht.

[Venkataraman et al., 2003] Venkataraman, A., Ferrer, L., Stolcke, A., and Shriberg, E. (2003). Training a prosody-based dialog act tagger from unlabeled data. *Proc. of the IEEE ICASSP*.

[Venkataraman et al., 2005] Venkataraman, A., Liu, Y., Shriberg, E., and A., S. (2005). Does active learning help automatic dialog act taggin in meeting data. In *Proc. Eurospeech*.

[Venkataraman et al., 2002] Venkataraman, A., Stolcke, A., and Shirberg, E. (2002). Automatic dialog act labeling with minimal supervision. In *Proceedings of the 9th Australian International Conference on Speech Science & Technology*.

[Verbree, 2006] Verbree, A. (2006). On the structuring of discussion transcripts based on utterances automatically classified. Master's thesis, University of Twente.

[Walton, 1996] Walton, D. (1996). *Argument Structure, A pragmatic Theory*. University of Toronto Press.

[Warnke et al., 1997] Warnke, V., Kompe, R., Niemann, H., and Nöth, E. (1997). Integrated dialog act segmentation and classification using prosodic features and language models. In *Proc. 5th Europ. Conf. on Speech, Communication, and Technology*, pages 207–210. Eurospeech.

[Webb et al., 2005] Webb, N., Hepple, M., and Wilks, Y. (2005). Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*.

[Wellner et al., 2004] Wellner, P., Flynn, M., and Guillemot, M. (2004). Browsing recorded meetings with ferret. In *In Proceedings of MLMI'04*. Springer-Verlag.

[Wellner et al., 2005] Wellner, P., Flynn, M., Tucker, S., and Whittaker, S. (2005). 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. ACM Press.

[Witten and Frank, 2000] Witten, I. H. and Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, San Francisco, CA.

[Xu et al., 2005] Xu, W., Carletta, J., Kilgour, J., and Karaiskos, V. (2005). Coding instructions for topic segmentation of the amimeeting corpus (version 1.1). http://www.idiap.ch/amicorpus/documentations.

[Yoshimi, 2004] Yoshimi, J. (2004). The structure of debate. Technical report, University of Claifornia, Merced.

[Yu and Seide, 2005] Yu, P. and Seide, F. (2005). Fast two-stage vocabulary independent search in spontaneous speech. In *Proc. ICASSP 2005*, Philadelphia.

[Zechner and Waibel, 2000] Zechner, K. and Waibel, A. (May 2000). Minimizing word error rate in textual summaries of spoken language. In *Proceedings of NAACL 2000, Seattle, WA, USA*.

[Zhang et al., 2002] Zhang, T., Damerau, F., and Johnson, D. (2002). Text chunking based on a generalization of winnow. *Journal of Machine Learning Research*, 2:615–637.

[Zimmermann et al., 2006a] Zimmermann, M., Hakkani-Tür, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., and Liu, Y. (2006a). The ICSI+ multilingual sentence segmentation system. In *International Conference on Spoken Language Processing, ICSLP 2006*, Pittsburg, USA.

[Zimmermann et al., 2005] Zimmermann, M., Liu, Y., Shriberg, E., and Stolcke, A. (2005). A* based joint segmentation and classification of dialog acts in multiparty meetings. In *Proc. 9th IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 215–219, San Juan, Puerto Rico.

[Zimmermann et al., 2006b] Zimmermann, M., Liu, Y., Shriberg, E., and Stolcke, A. (2006b). Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Machine Learning for Multimodal Interaction: 2nd International Workshop, MLMI 2005*, pages 187–193. LNCS 3869, Springer.

[Zimmermann et al., 2006c] Zimmermann, M., Stolcke, A., and Shriberg, E. (2006c). Joint segmentation and classification of dialog acts in multi-party meetings. In *Proc. 31st ICASSP*, volume 1, pages 581–584, Toulouse, France.