AMIDA Augmented Multi-party Interaction with Distance Access `http://www.amidaproject.org/` Integrated

Project IST–033812 Funded under 6th FWP (Sixth Framework Programme) Action Line: IST-2005-2.5.7 Multimodal interfaces

Deliverable D5.4: WP5 Work in Year 2

Due date: 30/09/2008 **Submission date:** 03/11/2008
Project start date: 1/10/2006 **Duration:** 36 months
Lead Contractor: DFKI **Revision:** 1

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

D5.4: WP5 Work in Year 2

Abstract: This deliverable presents a concise description of the progress in multimodal analysis and structuring made in the second of three years of the AMIDA project. It covers a large number of research areas and the presentations assume access to additional publications and previous deliverables, in particular AMIDA D5.2. Research results reported include: dialog act classification and segmentation, disfluencies, subjectivity and sentiment recognition, decision detection, dominance estimation, summarization and video editing.

Contents

1 Introduction

This deliverable presents a concise description of the progress in multimodal analysis and structuring made in the second of three years of the AMIDA project. It covers a large number of research areas and the presentations assume access to additional publications and previous deliverables, in particular AMIDA D5.2. Research results reported include: dialog act classification and segmentation, disfluencies, subjectivity and sentiment recognition, decision detection, dominance estimation, summarization and video editing. The individual advances are briefly summarized in the next section.

The two major overall trends in the second year have been on the one hand improvent of established methods, including the move to remote scenarios and on the other hand the first implementations of modules that run below real-time, run on-line with low latencies and have APIs that allow the connection to the hub, as the basic middleware for the implementaiton of prototypes in the context of WP6.

1.1 Overview of Results

The first main advance in dialogue act recognition was the implementation of a recursive classification strategy that was needed for the on-line version of our algorithm: a low-latency first result with an accuracy of 54.88% is recursively improved to 58.78% while the total processing time remains below real time. The second result in dialogue act recognition was achieved by applying discriminative re-ranking to automatic DA recognition, postprocessing the output of the generative switching DBN DA recogniser with a static discriminative classifier based on *linear chain* Conditional Random Fields. We achieved improvements on all the transcription conditions and on all the evaluation metrics, with reduction of 5–12% absolute.

Work on dialogue act segmentation concentrated on word specific models for six words at the segment boundaries that cause most of the errors in our previous work. Prosodic word specific boundary models can improve accuracy with values ranging from 7% to 20%.

New work on disfluency detection and removal is based on a hybrid approach that removes 12 of the 15 classes we identified with a relative improvement of over 42% compared to the baseline.

In two sets of experiments on subjectivity and sentiment recognition we have workd on the combination of single source classifiers and achieved an F_1 of 67.1 in distinguishing subjective from non-subjective utterances and an F_1 of 89.9 in distinguishing positive subjective from negative subjective utterances.

In a task-based evaluation of our work on automatic decision detection, we showed that decision focused extractive summaries help users to more effictively complete the task and result in decision minutes of higher quality.

Investing various models, we found an SVM approach to work best in determining the most and the least dominant participants in the meeting with accuracies of 91% and 89% respectively. Using only Visual Focus of Attention, we achieved a best performance of 72% on the most-dominant task.

In spoken term detection, including out of vocabulary words, we have extended the use of phoneme 3-grams to variable length multigrams with various constraints and achieved an accuracy of 63.0%, compared to 51.4% for a baseline LVCSR approach.

We have improved extractive summarization efforts by extending the simple Maximum Marginal Relevance algorithm to a beam search version. We have also work on developing upper bounds for the problematic, but commonly used evaluation scores, i.e., a maximum ROUGE summary score and maximum weighted precision.

The online automatic video editing implementation has been extended with an interface and respective functionalites for connection to the Hub to allow an inclusion in AMIDA prototypes. Offline video editing has been enhanced by new features, including Global Motion and automatically detected slide changes on the projector's channel.

2 Dialog Acts

This section reports on our efforts to enhance dialog act (DA) segmentation and classification. We have also moved some of the components, notably those with sub-realtime complexity, to on-line versions. This entails more complex architectures, as the algorithms rely on surrounding DAs as contextual features for maximum quality. As such context is not immediately available in an on-line version, we have developed an iterative architecture that provides initial results with minimum latency and improves the results in up to 60 iterations. A demonstration system has been developed that shows this implementation in conjunction with the disfluency removal, see section 3.

2.1 On-line Dialogue Act Classification

Previous research showed that the information of sourrounding DAs help a machine learning classifier to label the current segment. While the information of previous and upcoming DA labels are available in an offline development scenario, their calculation gets more complex in an online scenario where information can change over time when more DAs get segmented. Therefore, we concentrated our research on an effective estimation of these so-called *dynamic* features.

We developed an any-time algorithm that returns a first guess for the label of the new DA and refines the labels when more information is available: If a new DA gets segmented and labeled, the algorithm checks if adjacent DAs change their label due to the new segment. If so, this information gets propageted to further DAs. As can be seen in table 1, we were able to increase the classification performance from 54.88% to 58.78%, using a window of 40 DAs as initial window for re-classification. Of course, this recursive classification of the DAs needs more time but still keeps below real-time.

window	accuracy [%]	worst latency [s]
0	54.88	2
10	56.83	76
20	57.27	108
40	58.78	128
80	58.78	272

Table 1: Results of recursive Classification

2.2 *AMIDA* D^3 - DialogueAct and Disfluency Demo

Presenting results from classification tasks like, e.g., the DA classification often focusses on a comparison of different evaluation metrics and resulting numbers. This can be an insufficient way, as people are perhaps not familiar with the used evaluation metrics or just do not believe that the gained results can be preserved in a non-development scenario. That's why we thought about the implementation of a program whose only purpose is the visualization of our work on two fields of DA classification and Disfluency detection. We called this program *AMIDA* D^3 where D^3 is an abbreviation for "**D**ialogueAct and **D**isfluency **D**emo." Figure 1 shows a screenshot of the running program. It is designed as a meeting (re-)player that reads out the words of a meeting from the corpus and streams

them into the application - preserving their timestamps in the meeting. At the same time it plays the synchronized audio files for all four channels.

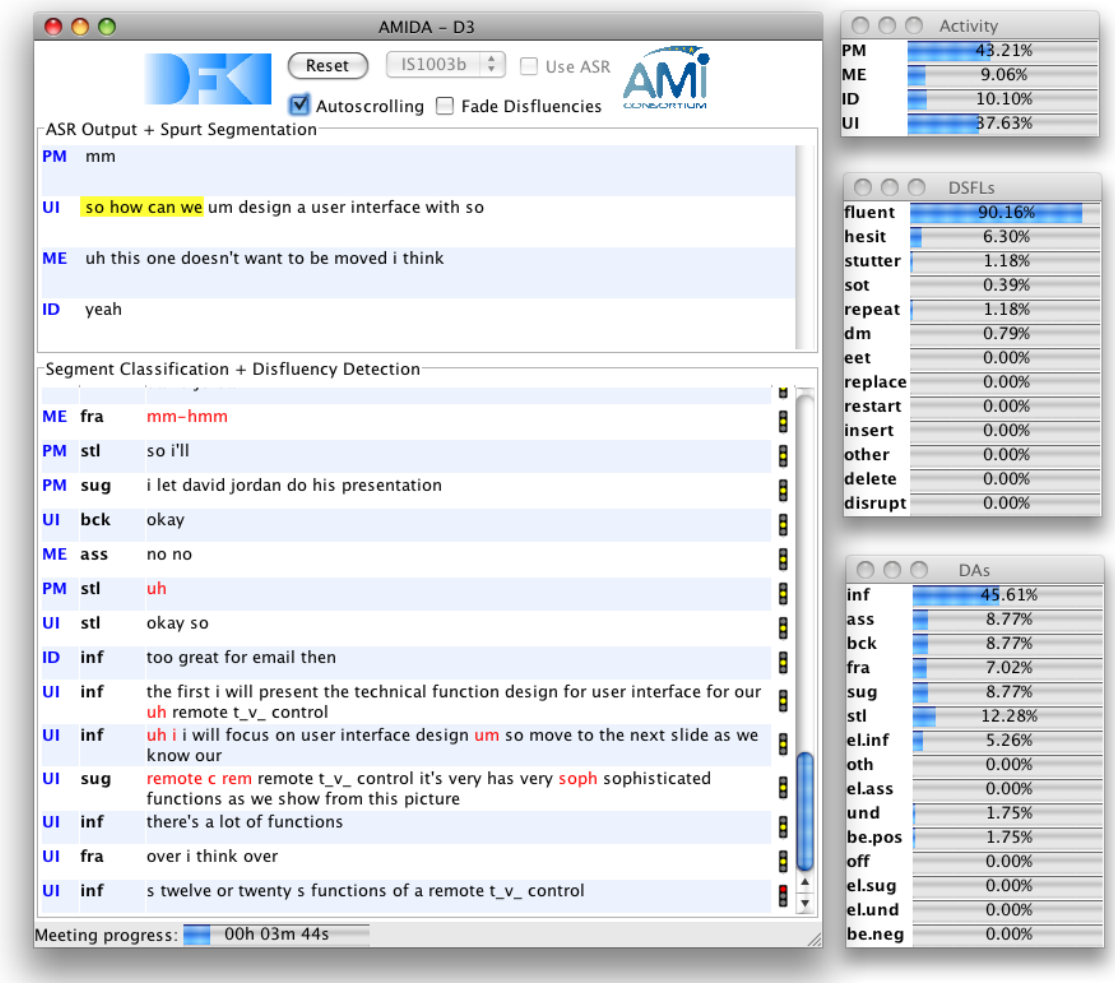

Figure 1: Screenshot of *AMIDA D*[3]

In the upper region which is called "ASR Output + Spurt Segmentation," each word gets streamed in its corresponding line, depending on who uttered it. If a segment is detected, the corresponding words are marked yellow and transferred to the lower part of the application which sorts the segments by their starting time in the meeting. Initially, the segment is unlabeled and no disfluency detection is performed. Hence, the traffic light in the right column is red. After some time[1], the disfluency detection is performed and words marked as disfluent are coloured either red or faded into light gray, depending on the users preferences. An additional thread is responsible for the on-line classification of the DAs. Thereby, the described algorithm from section 2.1 gets performed. The traffic light turns green as soon as a stable labeling of the DA segment is reached[2]. The three adjacent windows to the right show some optional statistics about the activity of each participant in the meeting, the distribution of the various types of disfluencies and the distribution of DA labels.

1. Time intervals are a variable parameter.
2. Our research showed that a label of a DA can be defined as being stable, if it is older than 60 segments

This demo has already been presented at different occasions, e.g., the MLMI and COI workshops, the CSP Summit, the Interspeech conference and various academic sites. The audiences repeatetly commented that it has an easy understandable structure which makes it obvious for the audience to understand what we are doing in our research.

2.2.1 Further work

Further steps will be to embed the presentation of at least one video stream in an extra window to support the multi-modality which is so far given by the play-back of the audio signal, aligned with the words.

2.3 Discriminative re-classification of the switching DBN recognition output

The adoption of discriminative classifiers such as Support Vector Machines (SVMs) [Vapnik, 1995] and Conditional Random Fields (CRFs) [Lafferty et al., 2001] to re-rank the output of sequential generative models has proven to be an effective technique in domains such as probabilistic parsing and statistical machine translation. For example in probabilistic parsing, a generative model estimates a list of parse hypotheses for each input sentence, then an additional discriminative model is used to rerank them [Collins, 2000, Koo and Collins, 2005]. In statistical machine translation a similar approach could be used to rerank n-best lists of candidate translations [Shen et al., 2004]. This technique may be applied to any preexisting system leaving it unaltered and exploiting temporal boundaries and recognition candidates estimated by the generative model. Moreover directly discriminant approaches explicitly optimise an error rate criterion, allowing the inclusion of new features with a limited computational overhead.

We have applied discriminative re-ranking to automatic DA recognition, postprocessing the output of the generative switching DBN DA recogniser outlined in Dielmann and Renals [2007, 2008] with a static discriminative classifier based on *linear chain* Conditional Random Fields [Lafferty et al., 2001]. A CRF classifier implemented with CRF++ [3] has been used to associate new DA labels with the best segmentations provided by the switching DBN. Five word related prosodic features (F0 mean, energy, word informativeness, word duration, pause length) were discretised and used in conjunction with the lexical information during the CRF re-labeling process.

Tables 3 and 2 report the recognition performances on the AMI 15 DA task before and after discriminative re-classification, respectively with and without the adoption of discretised prosodic features. The improvement is consistent on all the transcription conditions and on all the evaluation metrics, with reduction of 5–12% absolute. This improvement is mainly due to the discriminative use of the lexical content; the comparison between table 2 and 3 shows that prosodic features provide a marginal contribution of less than 0.5% on reference transcriptions, and 1.2% on fully automatic *ASR* transcriptions. This confirms that acoustics related features can help to discriminate between DA units with similar lexical realisations [Bhagat et al., 2003], but word identities play a more central role in DA classification.

3. Available from: `http://crfpp.sourceforge.net/`

Recognition metrics	Reference transcription	Automatic transcription
NIST-SU	59.3 (71.3)	71.8 (81.2)
DER	46.7 (51.9)	60.0 (64.1)
Strict	54.5 (62.1)	58.2 (64.7)
Lenient	36.5 (42.2)	41.7 (46.9)

Table 2: 15 classes AMI DA recognition error rates (%) of a CRF based re-classification system without the use of discretised prosodic features. Prior recognition performances using the generative switching DBN approach [Dielmann and Renals, 2008] have been reported in brackets.

Recognition metrics	Reference transcription	Automatic transcription
NIST-SU	59.2 (71.3)	71.3 (81.2)
DER	46.7 (51.9)	59.7 (64.1)
Strict	54.2 (62.1)	57.4 (64.7)
Lenient	36.0 (42.2)	40.5 (46.9)

Table 3: 15 classes AMI DA recognition error rates (%) of a CRF based re-classification system using lexical and prosodic features. Prior recognition performances using the generative switching DBN approach have been reported in brackets.

2.3.1 Discriminative re-classification using 4 broad DA categories

Additional DA recognition experiments were performed on the AMI corpus using a reduced number of DA categories. Early experiments of Hsueh and Moore [2007a,b] on automatic decision detection in conversational speech, suggested that replacing the 15 AMI DA classes with a reduced number of broader DA classes can improve decision detection. DA labels provide supporting evidence during the decision detection process, and are thus adopted as contextual features for a maximum entropy classifier. However not all the 15 labels play the same role on this task [Hsueh and Moore, 2007b]: stall and fragment DAs tend to precede or follow decision making segments; elicit type DAs precede and follow non decision making sentences; decisions are more frequent within inform and suggest DAs. Therefore it is reasonable to cluster together the DA types which provide similar cues. Following these considerations, the original 15 AMI DA classes can be grouped into a new set of 4 broad DA categories targeted on the automatic decision detection task. Table 4 shows the new 4 broad DA categories obtained by merging all DAs unrelated to specific speaker intentions (backchannel, stall, and fragment), by grouping information exchange DAs, forming a single class for elicit type DAs, and assigning all the remaining classes to a forth group.

The resulting 4 categories are unevenly distributed: information exchange accounts for more than half of the data, and elicit type DAs represent only 5.8% of the total number of DAs. Since the automatic mapping from 15 classes to 4 broad categories concerns only the DA labels but not their temporal segmentation, the original 15 DA manually annotated segmentation is preserved, thus both annotation schemes result in sharing the same segmentation.

Category	AMI DA classes	Proportion %
Category 1 *No speaker intention*	*backchannel* (17.6%) *stall* (6.3%) *fragment* (13.0%)	36.9
Category 2	*inform* (26.6%) *suggest* (7.5%) *assess* (16.7%)	50.8
Category 3 *Elicit classes*	*elicit inform* (3.4%) *elicit offer or suggestion* (0.5%) *elicit assessment* (1.7%) *elicit comment understanding* (0.2%)	5.8
Category 4 *Other classes*	*offer* (1.2%) *comment about understanding* (1.8%) *be positive* (1.8%) *be negative* (0.1%) *other* (1.8%)	6.7

Table 4: Four broad Dialogue Act categories obtained by merging the fifteen specialised AMI DA classes, with the percentage of DAs in each category.

The use of a Conditional Random Field static discriminative classifier to re-estimate the output of a joint generative DA recogniser proved to be effective on the 15 AMI DA task. This approach can be similarly applied to estimate a new classification in terms of 4 broad categories, starting from the 15 DA recognition output provided by the switching DBN model. A linear chain CRF, trained on discretised prosodic features and on word identities, can be used to associate DA labels drawn from the dictionary of 4 broad DA categories to the best segmentation output provided by the switching DBN.

Recognition metrics	Reference transcription	Automatic transcription
NIST-SU	42.5 (51.7)	53.9 (62.1)
DER	33.2 (38.8)	44.8 (51.9)
Strict	39.3 (47.7)	39.8 (51.1)
Lenient	13.1 (17.8)	15.6 (21.5)

Table 5: 4 classes DA recognition error rates (%) of the CRF based re-classification applied to the output of switching DBN model targeted on 15 classes. Best recognition performances using a switching DBN trained "from scratch" on 4 DA classes have been reported in brackets.

Table 5 reports the recognition performances achieved following this procedure. The best segmentation obtained using the 15 DA classes interpolated FLM setup has been re-classified using a linear CRF trained on 4 DA categories. A consistent improvement over a switching DBN system trained on 4 classes (results reported in brackets) can be observed on all the evaluation metrics, yielding an absolute reduction in the range of 2–10% according to the recognition metric.

DA segmentation using a switching DBN targeted on 15 classes, followed by CRF based

re-classification using just 4 categories, provided good performances on the AMI 4 broad DA recognition task. Moreover this approach allows to quickly reestimate the automatic DA classification output adapting it to a new DA tagset.

2.4 Work on ICSI meeting corpus

We have continued work on the hand-annotated Meeting Recorder Dialog Act (MRDA) corpus developed at ICSI. Weka classifiers predicted dialog acts in this corpus from language and prosodic features. We used a 5-way DA classification, as well as a cascaded classification approach consisting of a discrimination between long and short DAs, followed by several specialized classifiers. There was a particular emphasis on detection of questions and of broken-off utterances. Prosodic cues appeared to be useful, particularly when there were a significant number of examples of the target class for training.

2.5 Dialog Act Segmentation

In the previous deliverable of this work package -D5.2- we reported results on automatic dialogue act segmentation with models trained and tested on the AMI corpus. (See also the joint paper presented at MLMI 2008 op den Akker and Schulz [2008].) Detailed error-analysis revealed that there is a small number of frequently occurring words that causes most of the errors. The most important ones are: “and”, “yeah”, “I”, “so”, “okay”, and “but”.

Can we improve segmentation using word-specific models? Here, we report work we did and results we obtained in searching an answer to this question.

We used the speech analysis toolkit Praat[4] for analysing the AMI individual speaker’s audio files. Praat methods have been applied to a plot or graph drawn from the complete audio file, over a supplied time interval. Such an interval - in our case - is an audio sample with a start and end time corresponding to the start and end time encoded in the words XML file that corresponds to the current audio file. Thus, exactly one word. We can group all features that were extracted in 4 different types:

- Spectrum based
- Pitch based
- Intensity based
- Formant based

For each of the 6 words we trained separate prosodic models and tested them using a number of different classifiers from the WEKA toolkit. We started with an extensive search for the best prosodic features, based on experiences in the field (for references see op den Akker and Schulz [2008]). We used the information gain measure to see what features are most informative for the task, which is to tell whether the word-instance is or is not the start of a new dialogue act segment.

In the *initial stage* we trained word-specific models using *only prosodic features of the word itself*, that is: excluding the surrounding words or pauses before or after the word in the audio stream. At this stage we used the features on the word-instance’s time-intervals

4. www.praat.org

given in the AMI words layer (the forced alignment word boundaries), as we did in our previous research on dialogue act segmentation.

The results of these *initial* experiments can be seen in the left-most column of Tables 6 and 7. The values in the table are the percentages of correctly classified instances. It shows results obtained with a number of different types of classifiers - all using the same feature set. The evaluation methods used are either train/test split (90/10) or ten-fold cross-validation (cv). The names of the classifiers listed refer to the names used in the WEKA toolkit witten and Frank [2005]. ZeroR is the baseline, based on the most frequent occurring label in the data set. Notice that the data sets sometimes differ substantially in both size as well as in label distribution.

It is apparent that of the six words, there are three that roughly approximated the same error percentage as we saw for these words in our previous results, the value of *per* (previous error percentage) given in the tables, (as reported in op den Akker and Schulz [2008]). The other three word models performed even less satisfactory.

2.5.1 Re-alignment of word intervals

One of the Praat scripts encodes words with their start and end time to a textgrid file. Combining these textgrids with the corresponding audio files is the base for the scripted data extraction process. Fortunately Praat can also combine these two in a graphic manner in which the spoken text is placed directly under the wave form displaying the audio signal. Furthermore there is the possibility of displaying intensity, formants and pitch in an spectrograph. It turns out -as could be expected when looking into the word XML files- that the end time of most words is set at the begin time of the next, making the actual alignment sometimes rather poor. After all we are sampling the interval specified by the word. A good example is made visible in Figure 2, showing an instance of 'yeah', taken from meeting IS1000d, headset0. The actual interval, that Praat samples over due to the encoded start and end time is the colored section in the middle. In the waveform (top row) the markings indicate the amplitude recorded, normalized to the maximum amplitude in this selection. The second row displays energy levels over the frequency spectrum of $0\tilde{5}$ kHz, pitch in the blue (F0) and average intensity in the yellow. Thus, the actual utterance of the word is displayed in the upper (and second) row. We can easily see that the end time of the words (wave form) and sample end time (colored interval) clearly do not match. Obviously this clutters the measurements since word length, pauses and various mean values are included in the feature vector. Different "yeah's" , for example, in equal settings (both being a dialogue act boundary and uttered in the same manner) could still vary greatly in values for a great deal of features simply because the next word was spoken much later for one occurrence than for the other. In such case, only the value of a feature "pause-after-word" should vary.

To be able to take feature values of a "clean" word interval an algorithm was used that measures intensity and adjusts the end time accordingly. This is done in the following steps:

1. Maximum intensity (i_{max}) is measured for the entire interval
2. Mean intensity (i_{mean}) is measured for the entire interval
3. The difference of these intensities (i_{delta}) is calculated

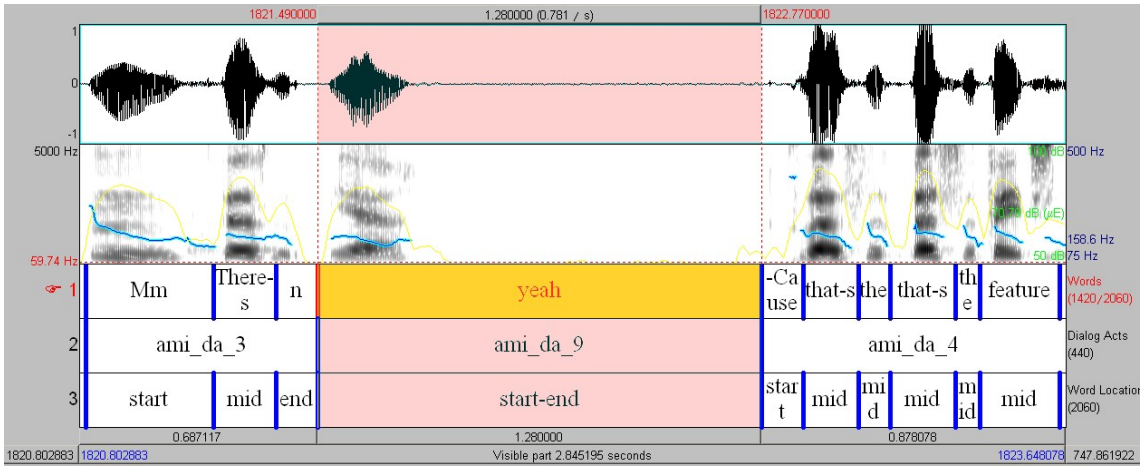

Figure 2: Begin and end times before re-alignment

4. If i_delta is at least 5 dB, a threshold value (i_{thresh}) for the intensity is calculated by $i_{mean} - i_{delta}$.
5. The end time of the interval (t_{end}) is sampled and the intensity (the mean over the sampling time) is measured (i_{end}).
6. If i_{end} is smaller than i_{thresh}, then t_{end} is lowered with 0.05 seconds.
7. Steps 5 - 6 are repeated until the sampled end time intensity i_{end} is equal to or greater than the threshold.
8. The new end time is set at t_{end} + 0.05 seconds.

When this algorithm is used on the *'yeah'* instance depicted in Figure 2 we get the following results.

```
The boundary times are: 1821.49 -- 1822.77
max intensity = 82.19
mean = 70.71
thus threshold set to 59.23
sampling boundary: 1822.77 --> energy level: 53.31
sampling boundary: 1822.72 --> energy level: 57.85
.
18 intermediate  steps of 0.05 seconds are left out
.
sampling boundary: 1821.77 --> energy level: 55.12
sampling boundary: 1821.72 --> energy level: 72.49
Done! The boundary times are: 1821.49 -- 1821.77
```

Figure 3 shows the same word instance as Figure 2. Again the colored section indicates the actual interval. (In the image it is selected by hand, however the times listed in above listed output: 1821.49 - 1821.77 correspond to the times left and right of the selection in the top of the figure.) We see that, although not matching completely, the correspondence of new start and end time to the actual sound has greatly improved. About 40% of the word instances alignments have actually been changed by the algorithm.

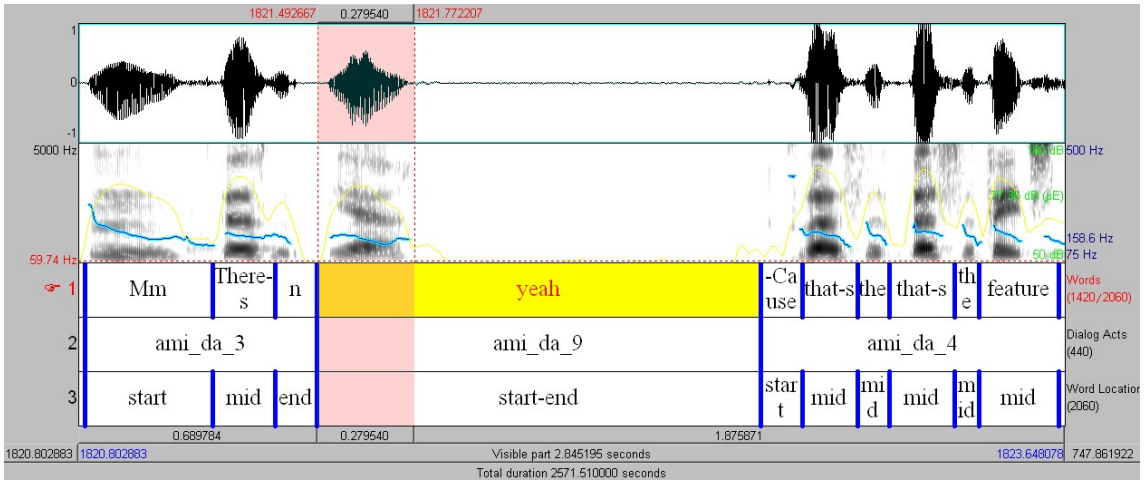

Figure 3: Begin and end times after re-alignment

2.5.2 Gender Specific Word Models

We repeated the initial experiments but now using the re-aligned word intervals. Moreover, we included three extra features. The first two extra features encode the *pause length before* and the *pause length after* the word. A third feature encodes speaker's *gender*. The results of the experiments are depicted in the third column (headed *"Both genders"*) of Tables 6 and 7. We also trained and tested *gender specific models*, i.e. after splitting the data for each word type into data of male and data of female speakers we removed the -now obsolete- gender feature in the latter models. The results on the male and female data sets are shown in the last two columns of Tables 6 and 7.

From the results we can conclude that:

- After re-alignment, for some words (see "and", "yeah") prosodic word specific models can be build to predict whether the word is a dialogue act boundary or not with an accuracy that improves the baseline accuracy significantly with values ranging from 7% to 20%.
- It is not clear if such word specific prosodic models exist for all the "problematic" words for dialogue act segmentation. See the results for the word "but" in Table 7.
- For some word types word specific prosodic models trained on gender specific data perform better than models that include a gender feature and that are trained on the whole data set. See the results obtained for the word "but" and the word "and". For some word types however, the models trained on the whole set perform as good as at least one of the gender specific models. See the results for the words "I" and "so".
- There is no single type of classifier that consistently performs better than other types, but the best results are often obtained with the SMO classifiers.

Whatever the results, in any case the re-alignment procedure produces more realistic word alignments, improving the validity of the results obtained. Table 8 lists the most relevant features measured by information gain.

And (per=35)		N=14581; Male=9790; Female=4791		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	49.3	52.7	52.9	52.8
bayesnet(90/10)	64.9	68.5	70.3	67.9
naiveBayes(90/10)	63.7	70.3	72.2	70.4
naiveBayes(cv)	64.5	70.7	71.1	69.9
J48(90/10)	62.9	68.1	67.4	67.3
SMO(75/25)	-	71.5	73.1	71.9
Yeah (per=20)		N=18760; Male=13884; Female=4876		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	76.8	77.8	75.7	78.1
bayesnet(90/10)	74.9	87.0	86.2	85.0
naiveBayes(90/10)	75.7	87.5	85.7	87.1
naiveSimple(90/10)	71.4	87.7	86.2	87.1
J48(90/10)	77.7	86.6	88.2	86.5
SMO(75/25)	-	88.5	88.4	87.9
I (per=22)		N=14173; Male=10089; Female=4084		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	64.2	64.8	64.7	59.2
bayesnet(90/10)	63.6	70.5	66.8	70.9
naiveBayes(90/10)	63.0	75.8	72.4	70.7
naiveSimple(90/10)	-	72.4	73.0	70.8
J48(90/10)	66.2	74.5	75.4	72.1
SMO(75/25)	-	-	-	75.1

Table 6: Classification results of the Initial (before re-alignment) experiments (second column) and for the experiments after re-alignment on three different data sets for each word: both genders, males and females apart (part I).

So (per=28)		N=10251; Male=7451; Female=2800		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	52.9	67.3	67.8	62.9
bayesnet(90/10)	43.1	69.6	71.6	71.4
naiveBayes(90/10)	43.6	67.9	71.8	65.7
naiveSimple(90/10)	44.0	68.2	70.0	-
J48(90/10)	42.4	69.9	72.8	68.9
SMO(75/25)	-	73.5	74.0	72.5
Okay (per=31)		N=7443; Male=5280; Female=2163		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	76.8	74.6	71.4	80.6
bayesnet(90/10)	69.0	84.2	78.8	86.6
naiveBayes(90/10)	63.6	81.9	78.8	81.6
naiveSimple(90/10)	43.1	77.0	78.0	80.6
J48(90/10)	77.2	85.9	83.0	81.6
SMO(75/25)	-	85.5	87.3	88.0
But (per=33)		N=6110; Male=4665; Female=1445		
Classifier	Initial Results	Both genders	Male	Female
zeroR(90/10)	68.4	72.7	66.6	73.8
bayesnet(90/10)	67.3	67.4	69.0	73.8
naiveBayes(90/10)	64.6	61.2	64.9	62.1
naiveSimple(90/10)	59.6	57.9	62.3	60.0
J48(90/10)	68.6	67.8	69.0	65.5
SMO(75/25)	-	69.6	67.8	76.2

Table 7: Classification results of the Initial (before re-alignment) experiments (second column) and for the experiments after re-alignment on three different data sets for each word: both genders, males and females apart (part II).

weight	feature	weight	feature
0.3193923	t-pause-before	0.0220955	formant2-max-t
0.0734436	intensity-sd	0.0205103	formant2-max
0.0560866	intensity-min-t	0.0192019	pitch-sd
0.0497704	intensity-min	0.0187707	formant3-stdDev
0.0326918	pitch-voiced-fr-ratio	0.0151675	pitch-max-t
0.0298274	pitch-frames	0.014584	t-pause-after
0.0293294	t-length	0.0144478	formant3-max-t
0.0288881	formant1-min-t	0.0142481	formant2-stdDev
0.0277069	formant1-stdDev	0.0135864	formant2-min-t
0.0260166	intensity-max-t	0.0120058	formant1-min
0.0244067	formant3-max	0.0112442	pitch-voiced
0.0236667	formant1-max	0.0112333	formant3-mean
0.0231201	sp-bin-width	0.010174	pitch-max
0.0231201	sp-bins	0.0098765	pitch-min-t

Table 8: The 28 most informative features for dialogue act segmentation

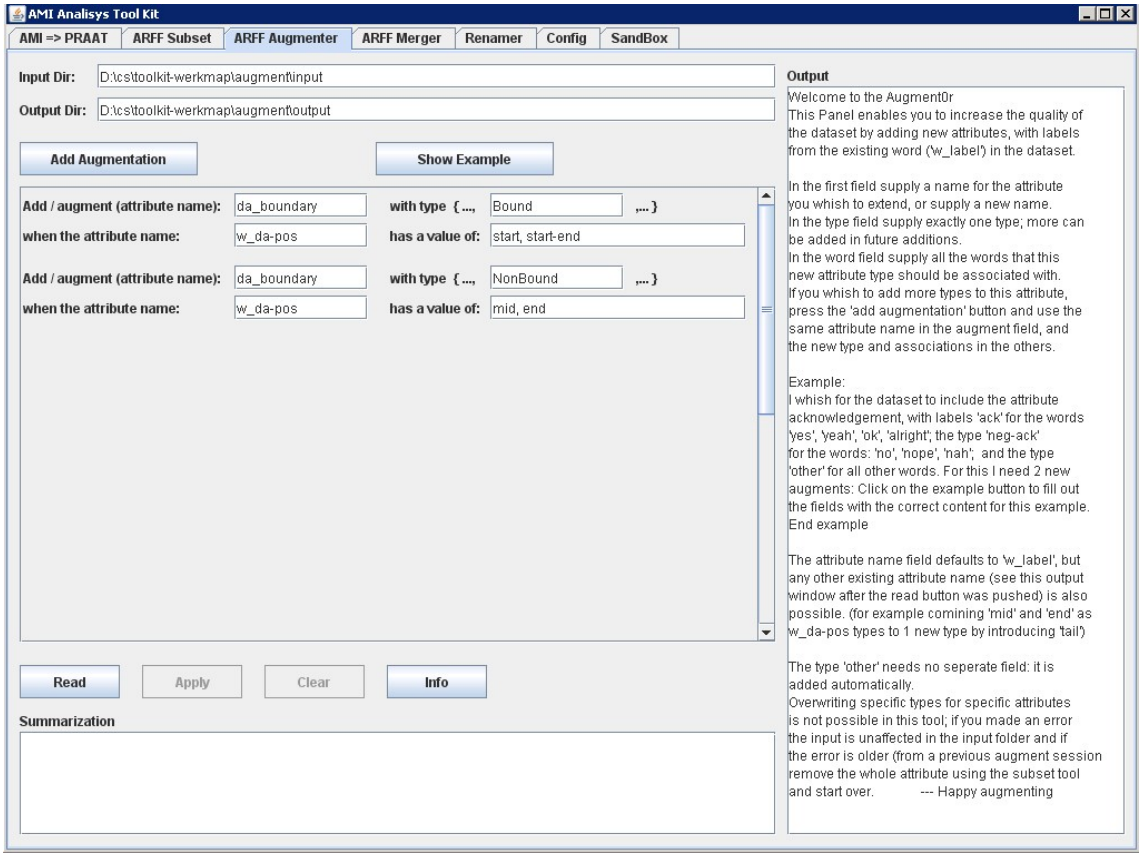

Figure 4: The GUI of the Prosodic Feature Extraction Toolkit

2.5.3 The Prosodic Feature Extraction Toolkit

The analysis performed for this research consists of a number of steps, leading from the input consisting of a number of NXT formatted annotation layers (word, dialogue act), and the audio signal files to a set of data-files in arff format suitable for further processing by the weka tools. A large number of Praat scripts have been written to extract the various feature values. For ease of processing the processing steps and scripts are collected into a Prosodic Feature Extraction Toolkit with a graphical user interface (in Java). The GUI of the toolkit is shown in Figure 4.

2.5.4 Future work

The feature extraction and classification method presented here are applicable in an on-line speech processing module. The word-specific models have to be integrated in a full on-line and efficient dialogue act segmentation (and labeling) system.

3 Disfluencies

3.1 Introduction

One main difference between written and spoken language are speech disfluencies. These are defined as "syntactical and grammatical [speech] errors" Besser [2006] and are mostly based on the incrementality of human speech production Ferreira et al. [2004]. In fact, 5% - 15% of spontaneous speech is disfluent and while human beings can filter these errors instantly, natural language processing systems often get confused by them. Therefore, an automatic system which separates the fluent speech material from the disfluent is highly desirable.

The scheme of the disfluency types this study is based on was developed earlier in the AMI project by Besser [2006] and contains a fine-grained annotation scheme of 15 different disfluency types (see table 9). Meanwhile, 45 meetings of the AMI and AMIDA corpora have been annotated with our disfluency scheme and we split the data in 80% training data and 20% evaluation data.

type	abbrev.	example
Hesitation	hesit	This **uh** is an example.
Stuttering	stutter	This is an **exa** example.
Disruption	disrupt	This is an example **and I**
Slip Of the Tongue	sot	This is an **y** example.
Discourse Marker	dm	**Well,** this is an example.
Explicit Editing Term	eet	This is **uh** this is an example.
Deletion	delete	**This really is** this is an example.
Insertion	insert	**This an** this is an example.
Repetition	repeat	**This is** this is an example.
Replacement	replace	**This was** this is an example.
Restart	restart	**We should,** this is an example.
Mistake	mistake	This **be** an example.
Order	order	This **an is** example.
Omission	omiss	This is **[ ]** example.
Other	other	

Table 9: Overview of all Disfluencies used in this study

3.2 Hybrid Detection System

A thorough investigation of our corpus and the disfluency scheme used showed a heterogeneity with respect to how the different disfluencies can be detected. This led us to the following design: Easily detectable disfluencies should be identified by a simple rule-based approach while the remaining disfluencies need a more sophisticated machine learning approach. Additionally, the disfluencies of the *Uncorrected* group cannot be detected through standard classification approaches as most of their material is missing in the speech and hence we decided to use a statistical N-gram based approach to cope with them. Furthermore, the usage of different detection techniques, each specialized and fine-tuned on its own disfluency domain, yields the advantage of an improved performance in conjunction with a reduced computational overhead at the same time.

3.2.1 Pattern Matching Approach

Our study showed that *Hesitations*, *Stutterings* and *Repetitions* are the only classes that are well suited for being recognized by lexical rules.

The detection of *Hesitations* is easy in the way that the top five of all *Hesitations* cover more than 98% of them. This means that detecting *Hesitations* is just a word-based matching of these identified words which are in fact: [*uh*, *um*, *mm*, *hmm*, *mm-hmm*]

Stutterings are detected with an algorithm that checks if the current word is "similar" to the beginning of the next word. This is done by counting the number of equal characters of the current and the next word divided by the length of the current word. If the resulting value exceeds the empirically measured threshold of 0.89 and both words are not equal, the algorithm identifies the current word as a *Stuttering*. Additionally, we check for "false-friends" which are words that fit into the described scheme even though they are fluent. To avoid matching them, these often appearing false-friends (in our study: [*we*, *no*, *on*, *so*, *it*]) are explicitly excluded from the detection.

Using regular expressions for the detection of *Repetitions* is an obvious approach and in fact leads to good detection results. Nevertheless, we had to adapt the expression `((?: \w+)+)\1 )` - which would result directly from their definition - to avoid a huge number of false positives and to decrease the processing time. First of all, we restricted the number of words we look for as it turned out that a length of 2 to 12 words for the whole disfluency is the best trade-off that we could find. Again, we explicitly exclude some words from the detection algorithm as they are common *Repetitions* that are assumed correct: [*very*, *okay*, *hmm*, *no*, *right*, *yes*, *one-nine*].

Furthermore, we are able to detect *Slip of the Tongues* with an inversed lexicon based approach which means that we derived a dictionary of fluent words from the corrected speech material of our training part and detect *Slip of the Tongues* as the words which are not included in this dictionary.

3.2.2 Machine Learning Approach

The machine learning approach is implemented with the help of the freely available WEKA toolkit Witten and Frank [2005] which contains many state-of-the-art machine learning algorithms and a variety of evaluation metrics. Furthermore, it allows to adapt other algorithms due to its simple interface.

We used machine learning based techniques to detect the following disfluency types: *Discourse Marker, Explicit Editing Term, Restart, Replacement, Insertion, Deletion, Disruption* and *Other*. We had to separate the detection of the *Disruption* class from the remaining ones as it needs a completely different feature set. The *Disruptions* are, according to their definition, classified as a complete segment while the remaining classes are detected word-by-word.

For both machine learning tasks, we trained and evaluated several algorithms to find the most suitable one for the task of the disfluency detection. In fact, the **Decision Tree (C4.5)** implementation of the WEKA toolkit outperformed all other algorithms in accuracy, F-Score and detection time but needs a lot of computation time for the training process.

All results are presented in section 3.3. We used four different types of features: **lexical, prosodic, speaker-related** and **dynamic**.

Lexical features are estimated on the word-layer and consider also the Part-of-Speech (POS) tags of the particular words. Next to the absolute words, we use some relative lexical features that describe the lexical parallelism between the current word to its neighbors.

As Shriberg et al. [1997] describes, **prosodic** features are well suited for the disfluency detection task and hence, we use them too. The term prosodic in this context means features that describe the *duration*, *energy*, *pitch* and *velocity* of the words. The *energy* and *pitch* values were normalized with a mean variance normalization per channel to reduce the influence of the microphones. Afterwards, we used these values to compute features like mean, variance and mode of the current word or segment and additionally, contextual features that described the prosodic parallelism of the surrounding elements.

The **speaker-related** features describe the speaker's role, gender, age and native-language as they appear in the corpus. These were used as we found a correlation between these characteristics and the rate of disfluent words.

The last type of features are **dynamic** features, that are generated during the process of the classification and describe the relationship between the disfluency type of the ongoing word to its neighbors.

3.2.3 N-gram based Approach

The detection of the *Uncorrected* disfluencies (*Omission, Mistake* and *Order*) was the most difficult task of this study because the speaker usually does not produce any explicit editing terms or any other information by making these errors. A statistical approach like the N-gram technique seemed to be a good way to gain information about the correctness of a word-order or a possible missing or superfluous word. We combined the probability of word-based N-grams and POS N-grams to gain more information about the correctness of the current word sequence and were able to calculate the difference between the "probability of the current sentence" to the particular alterations where, for example, two words get swapped or a word gets inserted.

N	OOV	PP
1	3.47%	1181.85
2	27.13%	2674.26
3	80.17%	33310.79

Table 10: N-gram Corpus Statistics

Unfortunately, this approach did not yield any detection improvements which is most likely due to the small size of the available corpus. The N-gram statistics have to be estimated on a huge text that must both be fluent and from the same context as the evaluation text. Both properties are fulfilled by the training set but it was too small to gain useful N-gram probabilities as seen in the perplexity and out-of-vocabulary values presented in table 10. Therefore, we excluded these three types of disfluencies from the detection part so far.

3.2.4 Hybrid Design

Figure 5 shows the schematic drawing of the architecture that has been developed. There we can see that the subsystems work on the data sequentially instead of a combined solution where the data gets process in parallel. In the first step, the rule-matching system processes the speech material. After that, the system's state advances to the machine learning approaches where the remaining types of the disfluencies are detected. If any disfluency was found, the speech material gets passed again into the rule-matching system. If not, the disfluency annotated stream gets processed by the *Disruption* detection system and after that made available for a possible subsequent NLP system.

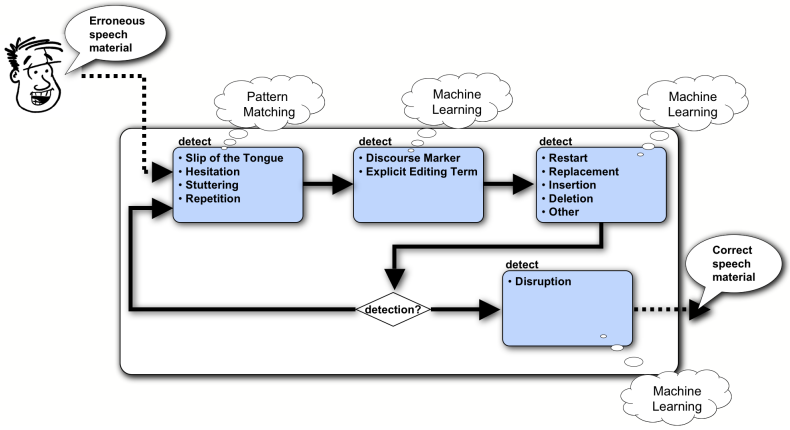

Figure 5: Design of hybrid Disfluency Detection System

The presented architecture emerged from a set of different design ideas that were all evaluated on the evaluation part of the corpus. The particular ideas differed in the way the subsystems were placed and in the way the speech was carried through them. In all design steps, we focussed our attention on keeping the precision as high as possible, because wrongly disfluent marked words have a worse influence on the meaning of the sentence than wrongly fluent marked ones.

3.3 Experimental Results

Since we combined the previously mentioned individual approaches, the hybrid approach is able to detect all their disfluency types[5]. In addition to the word-based evaluation metrics, we decided to compare the amount of disfluent dialogue acts with and without disfluency correction (see table 11), where a disfluent dialog act is defined as a dialog act that contains at least one disfluency. The baseline values are calculated by performing no disfluency detection at all and as only 14.1% of all words are disfluent, we gain a baseline of 85.9%. Table 11 shows that our hybrid approach is able to label 91.9% of all words correct which is a relative improvement of the accuracy of more than 42%. Furthermore, after cleaning the dialog acts of the found disfluencies, the amount of fluent dialog acts increased from 62.4% to 75.2%. The implemented system is very fast as it needs only 38 seconds processing time for the whole evaluation data. This results in a real-time factor of

5. Excluding the disfluencies of type *Omission*, *Mistake* and *Order*

0.0018. As there is a precision value for each single class and as presenting all these values would be too much, we decided to combine all single precision values by a weighted mean to one average value.

Word Level			**DA Level**		
	Baseline [%]	Result [%]		uncleaned [%]	cleaned [%]
Accuracy	85.9	91.9	fluent	62.4	75.2
avg. Precision	73.8	87.3	disfluent	37.6	34.8
Real Time	5:51 h				
Processing Time	38 sec				

Table 11: Performance of Disfluency Detection and Correction System

4 Subjectivity and Sentiment Recognition

Our work on recognizing subjective content in meetings has focused on two tasks: (1) recognizing subjectivity and sentiment using shallow linguistic features, and (2) investigating the use of prosody for detecting uncertainty.

4.1 Recognizing Subjectivity and Sentiment

An utterance may be subjective because the speaker is expressing an opinion, because the speaker is discussing someone else's opinion, or because the speaker is eliciting the opinion of someone else with a question. In addition to recognizing when an utterance is subjective, we also have worked on distinguishing between positive and negative subjective sentences (sentiment).

We approach the above tasks as supervised machine learning problems, with the specific goal of finding answers to the following research questions:

- Given a variety of information sources, such as text arising from (transcribed) speech, phoneme representations of the words in an utterance, and acoustic features extracted from the audio layer, which of these sources are particularly valuable for subjectivity analysis in multiparty conversation?
- Does the combination of these sources lead to further improvement?
- What are the optimal representations of these information sources in terms of feature design for a machine learning component?

In this research, we build on our work reported previously and continue to investigate the usefulness of shallow linguistic features, namely *character* and *phoneme n-grams* for subjectivity recognition.

4.1.1 Experiments

For this work we use 13 meetings from the AMI Meeting Corpus Carletta et al. [2005] annotated for subjective content using the AMIDA annotation scheme. For full details of the annotations as well as results for intercoder agreement, see Wilson [2008].

We conduct two sets of classification experiments. For the first set of experiments (Task 1), we automatically distinguish between subjective and non-subjective utterances. For the second set of experiments (Task 2), we focus on distinguishing between positive and negative subjective utterances. For both tasks, we use the manual dialogue act segments available as part of the AMI Corpus as the unit of classification. For Task 1, a segment is considered subjective if it overlaps with either a subjective utterance annotation or subjective question annotation. For Task 2, the segments being classified are those that overlap with positive or negative subjective utterances. For this task, we exclude segments that are both positive and negative. Although limiting the set of segments to be classified to just those that are positive or negative makes the task somewhat artificial, it also allows us to focus in on the performance of features specifically for this task.[6] We use 6226 subjective and 8707 non-subjective dialog acts for Task 1 (with an average duration of 1.9s, standard

6. In practice, this excludes about 7% of the positive/negative segments.

Table 12: Prosodic features used in experiments.

pitch	mean, std. dev, min, max, range, mean absolute slope
intensity (energy)	mean, std. dev, min, max, range, RMS energy
distribution energy in LTAS	slope, Hammerberg index, centre of gravity, skewness

deviation of 2.0s), and 3157 positive subjective and 1052 negative subjective dialog acts for Task 2 (average duration of 2.6s, standard deviation of 2.3s).

The experiments are performed using 13-fold cross validation. Each meeting constitutes a separate fold for testing, e.g., all the segments from meeting 1 make up the test set for fold 1. Then, for a given fold, the segments from the remaining 12 meetings are used for training and parameter tuning, with roughly a 85%, 7%, and 8% split between training, tuning, and testing sets for each fold. The assignment to training versus tuning set was random, with the only constraint being that a segment could only be in the tuning set for one fold of the data.

The experiments we perform involve two steps. First, we train and optimize a classifier for each type of feature using BoosTexter Schapire and Singer [2000] AdaBoost.MH. Then, we investigate the performance of all possible combinations of features using linear combinations of the individual feature classifiers.

4.1.2 Features

The two modalities that are investigated, prosodic, and textual, are represented by four different sets of features: prosody (PROS), word n-grams (WORDS), character n-grams (CHARS), and phoneme n-grams (PHONES).

Based on previous research on prosody modelling in a meeting context Wrede and Shriberg [2003] and on the literature in emotion research Banse and Scherer [1996] we extract PROS features that are mainly based on pitch, energy and the distribution of energy in the long-term averaged spectrum (LTAS) (see Table 12). These features are extracted at the word level and aggregated to the dialogue-act level by taking the average over the words per dialogue act. We then normalize the features per speaker per meeting by converting the raw feature values to z-scores ($z = (x - \mu)/\sigma$).

The textual features, WORDS and CHARS, and the PHONES features are based on a manual transcription of the speech. The PHONES were produced through dictionary lookup on the words in the reference transcription. Both CHARS and PHONES representations include word boundaries as informative tokens. The textual features for a given segment are simply all the WORDS/CHARS/PHONES in that segment. Selection of n-grams is performed by the learning algorithm.

4.1.3 Single Source Classifiers

We train four single source classifiers using BoosTexter, one for each type of feature. For the WORDS, CHARS, and PHONES, we optimize the classifier by performing a grid search over the parameter space, varying the number of rounds of boosting (100,

500, 1000, 2000, 5000), the length of the n-gram (1, 2, 3, 4, 5), and the type of n-gram. BoosTexter can be run with three different n-gram configurations: n-gram, s-gram, and f-gram. For the default configuration (n-gram), BoosTexter searches for n-grams up to length n. For example, if $n = 3$, BoosTexter will consider 1-grams, 2-grams, and 3-grams. For the s-gram configuration, BoosTexter will in addition consider sparse n-grams (i.e., n-grams containing wildcards), such as *the * idea*. For the f-gram configuration, BoosTexter will only consider n-grams of a maximum fixed length, e.g., if $n = 3$ BoosTexter will only consider 3-grams. For the PROS classifier, only the number of rounds of boosting was varied. The parameters are selected for each fold separately; the parameter set that produces the highest subjective F_1 score on the tuning set for Task 1, and the highest positive subjective F_1 score for Task 2, is used to train the final classifier for that fold.

4.1.4 Classifier combination

After the single source classifiers have been trained, they need to be combined into an aggregate classifier. To this end, apply a simple linear interpolation strategy. In the present binary class setting, BoosTexter produces two decision values, one for every class. For every individual single-source classifier (i.e., PROS, WORDS, CHARS and PHONES), separate weights are estimated that are applied to the decision values for the two classes produced by these classifiers. These weights express the relative importance of the single-source classifiers. The prediction of an aggregate classifier for a class c is then simply the sum of all weights for all participating single-source classifiers applied to the decision values these classifiers produce for this class. The class with the maximum score wins, just as in the simple non-aggregate case.

Formally, this linear interpolation strategy finds for n single-source classifiers n interpolation weights $\lambda_1, \ldots \lambda_n$ that minimize the empirical loss (measured by a loss function $\mathcal{L}$), with λ_j the weight of classifier j ($\lambda \in [0, 1]$), and $C_c^j(x_i)$ the decision value of class c produced by classifier j for datum x_i (a feature vector). The two classes are denoted with $0, 1$. The true class for datum x_i is denoted with $\hat{x}_i$. The loss function is in our case based on subjective F-measure (Task 1) or positive subjective F-measure (Task 2) measured on heldout development training and test data.

The aggregate prediction $\tilde{x}_i$ for datum x_i on the basis of n single-source classifiers then becomes

$$\tilde{x}_i = \arg\max_c \left(\sum_{j=1}^{n} \lambda_j \cdot C_{c=0}^j(x_i), \sum_{j=1}^{n} \lambda_j \cdot C_{c=1}^j(x_i) \right) \quad (1)$$

and the lambdas are defined as

$$\lambda_j^n = \arg\min_{\lambda_j^n \subset [0,1]} \sum_{i}^{k} \mathcal{L}(\hat{x}_i, \tilde{x}_i; \lambda_j, \ldots, \lambda_n) \quad (2)$$

The search process for these weights is implemented as a simple grid search over admissible ranges. In the experiments described below, we investigate all possible combinations of the four four different sets of features (PROS, WORDS, CHARS, and PHONES).

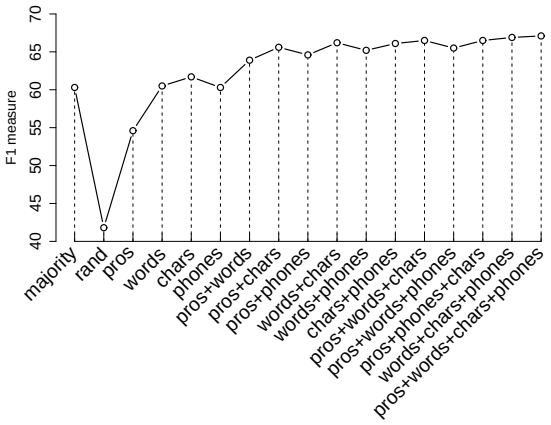

Figure 6: *Results Task 1: subjective* F_1

4.1.5 Results and Discussion

Results for the two tasks are given in Figures 6 and 7. We use two baselines, always choosing the subjective class (Task 1) or the positive class (Task 2), or a random choice based on the class disribution in the training data.

It is quite obvious that the combination of different sources of information is beneficial, and in general, the more information the better the results. The best performing classifier for Task 1 uses all the features, achieving a subjective F_1 of 67.1. For Task 2, the best performing classifier also uses all the features, although it does not perform significantly better than the classifier using only WORDS, CHARS, and PHONES. This classifier achieves a positive-subjective F_1 of 89.9.

Of the various feature types, prosody seems to be the least informative for both subjectivity and polarity classification. In addition to producing the single-source classifier with the lowest performance for both tasks, when added as an additional source of information, prosody is the least likely to yield significant improvements.

Throughout the experiments, adding an additional type of textual feature always yields higher results. The best performing of the features are the character *n*-grams. Of the single-source experiments, the character *n*-grams achieve the best performance, with significant improvements in F_1 over the other single-source classifiers for both Task 1 and Task 2. Also, adding character *n*-grams to other feature combinations always gives significant improvements in performance.

To verify that the above results were due to the classifier interpolation, we conducted two additional experiments. First, we investigated the performance of an uninterpolated combination of the four single-source classifiers. In essence, this combines the separate feature spaces without explicitly weighting them. Second, we investigated the results of training a single BoosTexter model using all the features, essentially merging all feature spaces into one agglomerate feature space. The results of these experiments confirms that interpolation outperforms both the unweighted and single-model combinations for both tasks.

For more details on the results from the experiments described in this section see Raaijmakers et al. [2008].

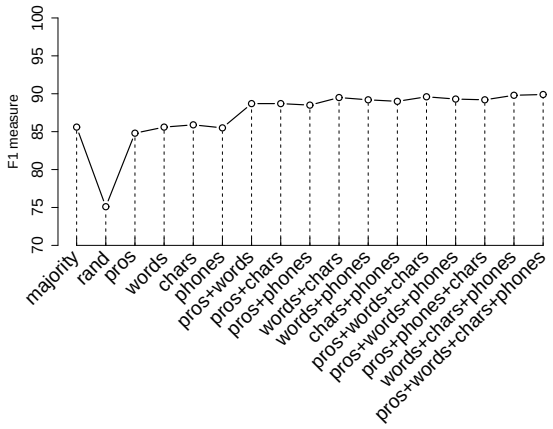

Figure 7: *Results Task 2: positive subjective* F_1

4.1.6 Ongoing and Future Work

The above experiments were performed using manual reference transcriptions as the source of data for words and phonemes, but we are in the process of replicating these exeriments using ASR and automatically recognized phonemes. We have investigated the performance of single-source classifiers for ASR words and characters and automatic phonemes. The phonemes for those experiments were produced by the real-time version of the Brno University of Technology phoneme recognizer Schwarz et al. [2004]. One goal of those experiments was to investigate how well *n*-grams from a fast by high-error phoneme recognizer would work for subjectivity recognition. Although performance was lowest for the automatic phoneme *n*-grams, they did they still significantly outperform the baseline, showing promise for moving toward phonemes for subjectivity detection in on-line and real-time systems. For more details on that work, see Wilson and Raaijmakers [2008].

In the near future, we will continue to incorporate new features into our subjectivity recognition systems, ranging linguistically motivated features to additional acoustic-prosodic and eventually visual features. We will continue to work towards real-time and on-line subjectivity recognition. Finally, we plan to work on recognizing not only subjective utterances but what the subjectivity is about. This is the next step toward building subjective content summaries of meetings, one of our ultimate goals.

4.2 Detecting Uncertainty

Our work on detecting uncertainty was focused on learning to automatically determine the confidence or self-conviction of speakers in multi-party conversation using prosodic features. Specifically, we were seeking answers to the following questions:

- Can prosodic features be used to automatically assess the degree of speaker (un)certainty un normal spoken dialogue?
- Which features, if any, qualify best as prosodic markers to the qualification of this (un)certainty?

The prosodic features evaluated include pitch, intensity, formant-related, and spectrum-related features.

4.2.1 Experiments

For these experiments we use 140 meetings from the AMI meeting corpus. The presence of lexical elements, specifically hedge words, was used to identify utterances with a high probability of being uncertain. The dialogue acts in the corpus were divided into three classes: *uncertain* (those that contain uncertainty hedges), *certain* (those that contain certainty hedges), and *neutral* (those that do not contain any hedges).

For the experiments, two datasets were created. The first used all of the dialogue acts containing uncertainty hedges (7,317) plus an equal number of randomly selected dialogue acts without hedges. The second dataset consisted of all dialogue acts with certainty markers (663) plus an equal number of dialogue acts with hedges.

Experiments both for distinguishing between hedging and not hedging and distinguishing between hedging and certainty were performed. Classification was performed using 10-fold cross validation, using J48 decision trees and Naive Bayes from the Weka toolkit witten and Frank [2005].

4.2.2 Results

For distinguishing between hedging and not hedging, performance for all prosodic features was 66% accuracy for decision trees and 68.5% accuracy for Naive Bayes. The best performing of the prosodic features were the formant-related and the pitch-related features, with the spectrum-related features also performing well for decision trees. For distinguishing between hedges and certainty, results for all prosodic features ranged from 56% for decision trees to 55.5% for naive bayes. For these experiments, which type of prosodic features performs the best varies between the algorithms. The spectrum related features perform the best for decision trees, but the format related features give the best performance for naive bayes.

For complete details on the above experiments and results, see Dral et al. [2008].

5 Decision Detection

In previous work, Hsueh and Moore [2007c], we developed a system for performing automatic decision detection in meetings. In this work, we conduct a task-based evaluation to investigate whether decision-focused summaries provide a benefit over general-purpose extractive summaries. The task asks a user to perform a "decision debriefing," with the goal of summarizing all the decisions made in a series of meetings.

5.1 Study Overview

For the study, we recruited 35 subjects (20 females and 15 males), aged 18-44) over a period of two months. Each subject first was asked to fill in a pre-questionaire about their prior experience in computer use and meeting attendance. This was followed by the experimenter demonstrating the meeting browser used in the experiment and explaining the basic task:

> "In 45 minutes or less, write a report to summarize the decisions made in the four meetings for upper management."

After the basic task explanation, the subject was given time to browse through a preselected meeting recording (distinct from the meetings used in the actual experiment) to familiarize herself with the browser interface.

When ready, subjects were given 45 minutes to complete the actual task. Throughout the session, a back-end component to the browser logged all user clicking and typing behavior. The user-generated decision minutes were logged as well.

At the end of each session, the experimenter asked the subjects to explain how they used the browser interface to find the decisions made in the meetings. Each subject also was asked to fill in a post-questionnaire about his or her perceived task success.

5.2 Meeting Browser Interface

The meeting browser used in this evaluation (see Figure 8) consists of three basic components: the audio-visual recording playback facility (top), the transcript display (lower left), and the extractive summary display (lower right). Each subject is equipped with headphones for listening to the audio of the meetings whenever necessary. Users can play the audio-video recording of a meeting from the beginning, or they can click on a particular extract in the summary window to be taken to that point in the meeting.

There are five tabs on the top of the brower interface: (1) the first four tabs take the users to each of the four meetings in the series, and (2) the last tab is the "writing tab," where users are asked to type in their summaries. Users can switch between these tabs at will.

5.3 Experiment Design

For this study, our research hypothesis is that a more succinct, decision-focused summary will help users to obtain an overview of meeting decisions more efficiently, to prepare meeting minutes more effectively, and to feel more confident in meeting preparation

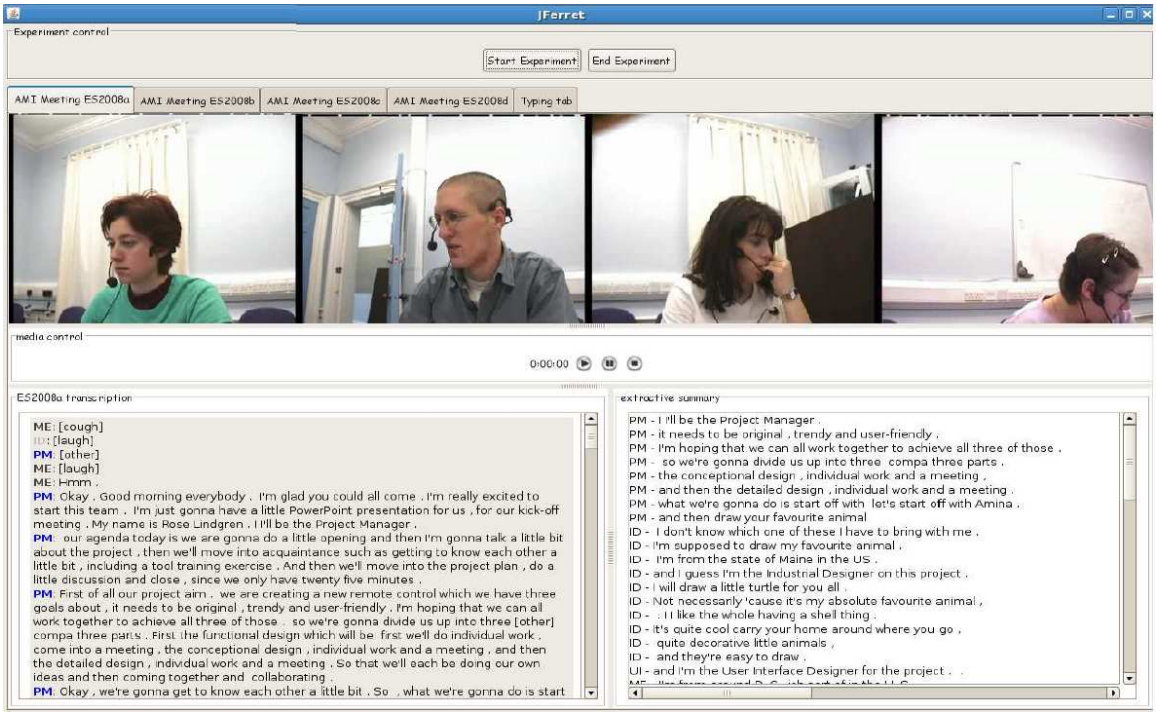

Figure 8: Meeting browser used in task-based evaluation

work. In addition to testing this hypothesis, we also evaluate the effect of automatic versus human transcriptions and automatic versus human annotated decision points.

The subjects in the study were randomly assigned to four groups. Each group was asked to accomplish the same task, but using different versions of the extractive summaries. Following are the four extractive summaries that were evaluated:

- Baseline (AE-ASR): automatic general purpose extracts, ASR transcription
- AD-ASR: automatic decision-focused extracts, ASR transcription
- AD-REF: automatic decision-focused extracts, manual transcription
- Topline (MD-REF): manual decision-focused extracts, manual transcription

5.4 Results

We evaluated the different browsers used in the study based on three criteria: 1) task effectiveness, 2) report quality, and 3) user percieved success. Task effectiveness was measured by comparing the user-generated decision minutes to the gold-standard decision points for the meeting series used in the study. For report quality, different aspects of the user-generated decision minutes are rated on 7-point Likert scales, including overall quality, completeness, conciseness, task comprehension, effort spent in writing, trustworthiness, and writing style. User preceived success looks at the level of perceived success and usability reported in the post-questionairres.

Figure 9 shows the results for task effectiveness across the four types of extractive summaries. The results show that decision-focused extractive summaries do help users to more effectively complete the task than the general-purpose extractive summaries. Decision-focused summaries also result in decision minutes of higher quality. In addition, the post-

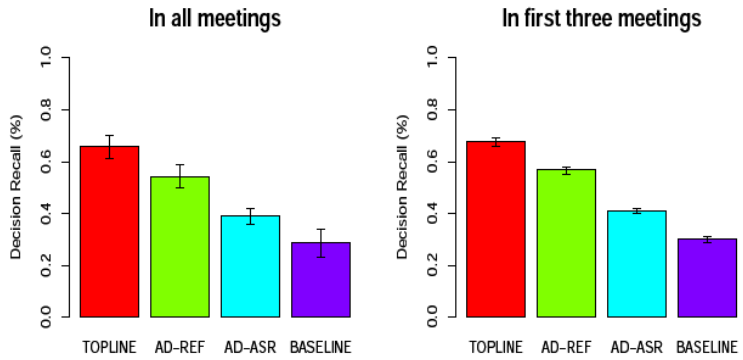

Figure 9: Task effectiveness as the average ratio of the decisions that are correctly found by the subjects

questionnaires reveal that the decision-focused summaries are perceived as easier to use, less demanding in the amount of effort required, and more efficient for the retrieval of relevant information.

From our evaluation of the effect of automatic versus human transcriptions and automatic versus human annotated decision points, we have the following main findings. First, although error-prone, the automatically generated decision-focused summaries do still assist users in producing high quality decision minutes and in feeling confident about their performance on the decision debriefing task. However, as compared to the topline summaries, using the automatic decision-focused summaries does result in a reduction in the number of identified decision points. Second, the ASR transcription does have negative effect on the task effectiveness and the quality of the decision minutes. It also increases the level of user-perceived pressure.

More details from this study are available in Hsueh and Moore [In submission].

6 Dominance

6.1 Targeted Objectives and Summary of Achievements

The overall goal of our work in dominance modeling this year was to continue investigating methods to estimate dominant behaviour. In addition, work was started to study the robustness of correlating noisy estimates of both speaking and visual activity. In summary, our work produced the following achievements:

- Investigation of unsupervised and supervised models for dominance estimation. From the human annotations of dominance, we defined 4 different tasks; classifying the most and least dominant person and with different variations in the perceived dominance by the annotators. Different models were tried for each task. The best achieved classification accuracy was 91% and 89% for the most-dominant and the least-dominant tasks using an SVM approach. In addition, we also conducted experiments into estimating the dominant clique in a meeting where the best accuracy was 80.8%.
- Investigation of estimating the most dominant person using visual focus of attention (VFOA). A few different measures using the VFOA of each participant were studied to investigate more complex features. A best performance of 71.9% was achieved.
- Investigation of the association of audio and visual streams from a single microphone and personal close-view cameras for dominance estimation. We developed an ordered association approach so that the most likely audio-video streams are associated first. Our best results showed that it was possible to associate 2 out of the 4 participants in a meeting 100% of the time. We also found that in 86% of the meetings, the most dominant speaker was correctly associated with their corresponding video channel.

6.2 Investigation of Unsupervised and Supervised Models for Dominance

The investigation of supervised and unsupervised models of dominance was concluded in the last period. In summary, the work carried out involved defining tasks to capture both extreme behaviour types (most and least dominant) and variability in the human annotations of dominance. Each meeting was 5 minutes long. Various audio and video features were extracted and both an unsupervised and supervised model was defined. For the unsupervised case, after features were extracted, they were accumulated over the meeting. The person with the highest value was chosen to be the most dominant, while the person with the lowest accumulated value was estimated as the least dominant. A supervised model was used by training a 2-class SVM classifier to identify the most dominant and non-dominant people in each meeting (and the same for the least dominant case). In addition, a task involving finding the dominant clique was also investigated, to see if there tended to be 2 more dominant people in each meeting.

The best performing single feature with the investigated unsupervised models was the total speaking length, which produced a classification accuracy of 85% and 86% for the most-dominant and the least-dominant tasks, respectively. With a Support Vector Machine-based approach and combined features, the best achieved classification accuracy was 91% and 89% for the most-dominant and the least-dominant tasks, respectively.

In addition, we also conducted experiments into estimating the dominant clique in a meeting. Using our SVM -based approach, we could predict the dominant clique with 80.8% accuracy.

6.3 Estimating Visual Dominance

Dovidio and Ellyson [Dovidio and Ellyson, 1982] suggested that someone who receives more visual attention is perceived to be more dominant. The received visual focus of attention (RVA) is defined as the total number of times that a particular person is the visual focus of any of the other participants in the meeting. The total received visual attention (RVA) for each participant i and their corresponding Visual Focus of Attention (VFOA), $f_t = (f_t^1, .., f_t^{M_p})$ at time t is defined as

$$RVA^i = \sum_{t=1}^{T} \sum_{j=1, j\neq i}^{M_p} \delta(f_t^j - i), \;\; i = 1, \ldots, M_p \;\; , \tag{3}$$

where T is the number of frames, $f_t^j \in \{1, \ldots, M\}$ where M is the number of focus targets, M_p is the number of participants ($M > M_p$), and $\delta(.)$ is the delta function such that $\delta(f_t^j - i) = 1$ when $f_t^j = i$. In our data, the focus targets were defined as the three other participants, the slide screen, the whiteboard, and the table. The table label was assigned whenever a person looked at the table or an object on it. For all gaze directed at other locations, an 'unfocused' label was also defined. We also encoded the ability of each person to 'grab' visual attention by considering the RVA feature in terms of events rather than frames.

Dovidio and Ellyson also [Dovidio and Ellyson, 1982] defined the VDR between dyads as the proportion of time a person looks at the other while speaking divided by the time a person looks at the other while listening. It encodes the displayed dominance through either active or passive participation. We extend the VDR to a multi-party scenario ($MVDR$). The 'looking-while-speaking' feature is redefined as when a person who is speaking looks at any participant rather than at other objects in the meeting. Similarly, the 'looking-while-listening' case involves actively looking at any speaking participant while listening. The $MVDR$ for person i is:

$$MVDR^i = \frac{MVDR_N^i}{MVDR_D^i} \;\; , \tag{4}$$

where the time that each participant spends looking at others while speaking is defined as

$$MVDR_N^i = \sum_{t=1}^{T} s_t^i \sum_{j=1, j\neq i}^{M_p} \delta(f_t^i - j) \;\; , i = 1, \ldots, M_p \;\; , \tag{5}$$

s_t^i is a binary vector containing the speaking status of each participant (speaking: 1, silence: 0). The time spent looking at a speaker while listening (i.e. not speaking) is defined as

$$MVDR_D^i = \sum_{t=1}^{T}(1 - s_t^i) \sum_{j=1, j\neq i}^{M_p} \delta(f_t^i - j)s_t^j. \quad (6)$$

Clearly, other definitions for the MVDR are also possible. It is important to note that we approximate listening as not speaking, while looking at a speaker. This accumulates all the frames where participant x_i is looking at all other x_j when x_i is silent and x_j is speaking.

6.3.1 Audio-visual Cue extraction

Speaking activity features are estimated as before. The visual focus of attention of each meeting participant was automatically estimated using the method described in the corresponding document of WP4.

6.3.2 Data and Annotations

The data and annotations are the same as those used and reported in the previous year.

6.3.3 Unsupervised most-dominant person classification

Using the *RVA* and *MDVR* measures defined in Eq.s 3 and 4 respectively, we estimated the most dominant person in each meeting for full and majority agreement data sets. We evaluated the performance of the *RVA* case for both frame and event-based cases. Also, to study the contribution of each element of the *MVDR*, we analyzed both the performance of the numerator and denominator separately as well as when combined. In each case except for the denominator, $MVDR_D$, defined in Eq.6, the person with the highest value was estimated to be the most dominant. For the case of the $MDVR_D$, the person with the smallest value was estimated to be the most dominant. The results for manual and automatically extracted cues are shown in Table 13.

	Meeting Classification Accuracy(%)			
	MostDom(Full)		**MostDom(Maj)**	
Method	**Manual**	**Auto**	**Manual**	**Auto**
RVA **(Time)**	58.8	67.6	52.6	61.4
RVA **(Events)**	70.6	38.2	61.4	42
MVDR	73.5	79.4	64.9	71.9
$MVDR_N$	79.4	70.6	70.1	63.2
$MVDR_D$	41.2	50	40.4	45.6
SL	85.3	85.3	77.2	77.2
Random	25			

Table 13: Percentage of correctly labeled meetings in the full and majority cases using manual and automatically estimated cues. SL: Speaking length.

Firstly, we considered the performance on the 34 meetings with full annotator agreement. We studied firstly the ideal case, where human annotations of speaking activity and VFOA were used. Using *RVA* events appeared to improve the performance compared to

time (from 58.8% to 70.6%). Interestingly, this feature was quite discriminative, using just visual cues. The introduction of speaking activity features with the *MVDR* appeared to improve the performance. Also, the *MVDR* did not seem to perform as well as just the using the $MVDR_N$, which performed the best at 79.4%. $MVDR_D$ had the worst performance of 41.2%.

Using the automated estimates, the best performing feature was the *MVDR* (79.4%). The *RVA* (Events) feature seemed to perform better in the manual rather than automatic case. This was probably because the estimates were smoothing out shorter events. The $MVDR_N$ feature seemed to perform worse compared to its manual counterpart. In contrast, the automatic case suggested a much better estimate using the $MVDR_D$ feature. In terms of the decrease and increase in performance between $MVDR_N$ and $MVDR_D$, respectively, when we compare the manual to automated versions, we observe that while the VFOA estimates of a speaker may not be affected by their own speaking activity, those of a listener are clearly conditioned on the conversational events.

Finally, analyzing the results using the manual and automated dominance estimation results in Table 13 for the majority agreement data-set, there was a consistent drop in performance while the relative differences between feature types and also manual and automatic labels were similar.

6.3.4 Conclusion

In this period, we have shown that extending Dovidio and Ellyson's measures of dominance to the group case was indeed effective. Our study also suggests that while audio cues are very strong, visual attention is also quite discriminant and could be useful in the absence of audio sensors. However, we have yet to discover other features that are jointly complementary. A more in-depth study of modifications to the VDR to the multiparty case is reserved for future work.

6.4 Associating Audio And Visual Streams From a Single Microphone and Personal Close-View Cameras for Dominance Estimation

If only a single microphone is available, speaker diarization must be used to identify the number of speakers and when they speak. From our previous work, reported in D5.2, we found that the speaking length worked best as a single cue for estimating the dominant person. However, once this information is extracted, we are only able to obtain a vocal sample of the dominant person, and not what they look like. By correlating the audio and visual streams together, we were able to identify each of the speakers in the meeting by a video channel, where it was assumed that each speaker was captured by their own close-view camera.

6.4.1 Audio-visual Cue extraction

For this work, speaker diarization was used to extract the number of speakers and when they spoke from a single audio source. For experimental reasons, we tried sources with an increasingly noisy SNR, and also different strategies for improving the diarization to see how these conditions affected the audio-visual association. For the video features, we

used the compressed-domain video features, which were reported previously, in order to obtain a motion activity value for each participant at each frame.

6.4.2 Associating Speaker Clusters with Unlabelled Video Channels

For each pair-wise combination of speaking and visual activity channels, their corresponding normalised correlation was calculated. We then matched the channels by using an ordered one-to-one mapping based on associating the best correlated channels first. Three different evaluation criteria were used to observe the differences in discriminability of the data by varying the leniency of the scoring into soft, medium and hard criteria : *EvS* gives each meeting a score of 1 if at least 1 of the 4 speech and visual activity channels match correctly; *EvM* scores 1 if at least two of the channels match correctly; *EvH* scores 1 only if all 4 visual activity channels are assigned correctly.

To evaluate the associations, we computed the pair-wise correlation between (*i*) the speaker clusters and visual activity features and (*ii*) the speaker clusters and the ground truth speaker segmentations. The mappings for both cases were calculated again based on an ordered one-to-one mapping starting from the pair with the highest correlation. If there were fewer speaker clusters than motion channels, mappings were forced to ensure each motion channel mapped to a speaker cluster. Finally, we integrated these association results back into the dominance task by checking all correct mappings to see if they matched with the longest cluster length (which we expect to be the most dominant person).

6.4.3 Results

The speech-visual activity association was performed on 21 5-minute segments where all the participants were always seated in their close-view camera. We tested using the speech activity output generated from different speaker diarization strategies and conditions described. Finally, we used both the ground truth segmentations and headset segmentations to evaluate the mappings. In all, 16 different combinations of evaluation criteria and reference segmentations were used and for each of these combinations, we had 24 different experimental conditions for the diarization output.

Table 14 shows the results using our 3 evaluation strategies and different reference speaker segmentations. A degradation in performance is observed as the evaluation criteria becomes more strict. The best average score was achieved by the *EvS* case with an average and highest performance of 93% and 100% respectively.

	EvH	*EvM*	*EvS*
Average	0.31	0.8	0.98
Max	0.48	1	1
Min	0.14	0.62	0.9

Table 14: Summary of the performance using all 4 performance evaluation strategies and either the ground truth or automatically generated headset speaker segmentations. The evaluation criteria are as described before.

Table 15 shows the percentage of meetings where the association of the longest speaker cluster with the correct visual activity channel was made. The best performing dominance

and association results were achieved by using the headset segmentations with an average performance of 70% where there were 3 cases where a performance of 86% was achieved.

Average	0.7
Max	0.86
Min	0.52

Table 15: Percentage of meetings where the correct mapping was given to the cluster with the longest speaking length.

7 Speech indexing and search

Spoken term detection (STD) is an important part of meeting processing. The most common way to perform STD is to use the output of a large vocabulary continuous speech recognizer (LVCSR). Rather than using the 1-best output of LVCSR, the state-of-the-art STD systems search terms in lattices - acyclic directed graphs of parallel hypotheses. In addition to better chances to find the searched term, the lattices also offer to estimate the confidence of given query hit.

A drawback of the LVCSR system is, that it recognizes only words which are in an LVCSR vocabulary, so that the following STD system can not detect out-of-vocabulary words (OOVs). Therefore, we are often recurring to sub-word units. The most intuitive sub-word units are phones. Another type of sub-word units are syllables, phone n-grams, multigrams or broad phonetic classes which are all based on phones.

7.1 Multigram based sub-word modeling

In our prior work at BUT Szoke et al. [2006], we have used sequences of overlapped 3-grams for search. However, words shorter than 3 phones must be processed in a different way or dropped. Another drawback of the fixed length is that the frequencies of units are not taken into account although some units are more frequent than others. **Variable length units** can be used to overcome this problem: a rare unit is split into more frequent shorter units while a frequent unit can be represented as a whole. The other advantage is that variable length units can reflect word sequences and compensate for missing word language model.

Disk space and computational requirements are also important from practical point of view – stored data and search time must be kept as small as possible. The decoding should be fast and an indexing must be used. The trade-off between index size and search accuracy must be also included to the evaluation.

Therefore, BUT has investigated the use of *multigrams* as sub-word units. We studied, which impact multigram parameters have on the accuracy and index size. We tried to find the optimal length and pruning of multigram units. We also proposed two improvements of multigram training algorithm to reflect word boundaries (see Table 16):

1. **disabling the silence in multigrams**, where the silence is considered a unit separating words which should not be part of sub-word units.
2. **not allowing multigrams to cross word-boundaries**

word	sil YEAH I MEAN IT IS sil INTERESTING									
xwrd	sil-y-eh-ax		ay-m-iy-n		ih-t-ih-z-sil			ih-n-t-ax-r	eh-s-t-ih-ng	
nosil	sil	y-eh-ax	ay-m-iy-n		ih-t-ih-z		sil	ih-n-t-ax-r	eh-s-t-ih-ng	
noxwrd	*sil*	*y-eh-ax*	*ay*	*m-iy-n*	*ih-t*	*ih-z*	*sil*	*ih-n	t-ax-r-eh-s	t-ih-ng*

Table 16: Examples of different multigram segmentations.

7.2 Experimental evaluation

The evaluation was done on the NIST STD06[7] development set (data and term list). The original NIST STD06 development term set for CTS contains a low number of OOVs. First of all, 124 terms containing true OOVs were omitted. Then, we selected 440 words from the term set and a further 440 words from the LVCSR vocabulary. A limited LVCSR system was created (denoted by *WRDRED* which means "reduced vocabulary") where these 880 words were omitted from the vocabulary. This system had reasonably high OOV rate on the NIST STD06 DevSet. The term set has 975 terms of which are 481 in vocabulary (IV) terms and 494 OOV terms (terms containing at least one OOV) for the reduced system. The number of occurrences of the IV terms is 4737 and 196 of OOV terms.

7.2.1 UBTWV - Upper Bound TWV

We used Term Weighted Value (TWV) for evaluation of spoken term detection (STD) accuracy of our experiments. The TWV was defined by NIST for STD 2006 evaluation. One drawback of TWV metric is its one global threshold for all terms. This is good for evaluation for end-user environment, but leads to uncertainty in comparison of different experimental setups. We do not know if the difference is caused by different systems or different normalization and global threshold estimation. This is a reason for the definition of *Upper Bound TWV* (UBTWV) which differs from TWV in individual threshold for each term. Ideal threshold for each term is found to maximize the term's TWV:

$$thr_{ideal}(term) = \arg\max_{thr} TWV(term, thr) \tag{7}$$

The UBTWV is then defined as

$$\begin{aligned} UBTWV = \quad & 1 - \underset{term}{average}\{p_{MISS}(term, thr_{ideal}(term)) + \\ & \beta p_{FA}(term, thr_{ideal}(term))\}, \end{aligned} \tag{8}$$

where β is 999.9. It is equivalent to a shift of each term to have the maximal $TWV(term)$ at threshold 0. Two systems can be compared by UBTWV without any influence of normalization and ideal threshold level estimation on the systems TWV score. The *UBTWV* was evaluated for the whole set of terms (denoted *UBTWV-ALL*), only for in-vocabulary subset (denoted *UBTWV-IV* and only for out-of-vocabulary subset (denoted *UBTWV-OOV*.

The other evaluation metrics were phoneme recognition accuracy and index size.

7.3 Results and conclusions

Table 17 compares word, phone and multigram based systems from phone and spoken term detection accuracy point of view. The *WRDREDwrd* was the LVCSR (with reduced vocabulary) on the word level (terms are word sequences). The *WRDREDphn* was the

7. 2006 NIST Spoken Term detection Evaluations, `http://www.nist.gov/speech/tests/std/2006/`

Unit	LM	Phone	UBTWV			SIZE
	n-gram	ACC	ALL	IV	OOV	
WRDREDwrd	2	-	51.4	73.4	0.00	0.56Mw
WRDREDphn	2	65.40	54.0	55.4	50.8	4.34Mp
phn-LnoOOV	3	59.66	48.3	45.3	55.2	6.38Mp
mgram-xwrd	3	65.25	55.9	55.2	57.7	1.4Mw/3.6Mp
mgram-nosil	3	65.42	58.4	57.8	59.7	1.2Mw/4.1Mp
mgram-noxwrd	3	65.10	**63.0**	64.7	59.3	1.7Mw/3.7Mp

Table 17: Comparison of word, phone and multigram systems.

LVCSR switched to phone level. The best phone accuracy was achieved by the *nosil* constrained multigrams. However, better STD accuracy was achieved by the *noxwrd* constrained multigrams. It is important to mention that multigram lattices are significantly smaller and the recognition network is approximately the same size compared to phones. Details of this work will be presented in Szoke et al. [2008].

8 Summarization

8.1 Abstractive summarization

To generate abstractive summaries, one of the key challenges is to come up with a model that is capable of representing summary contents. In theory, to create a comprehensive model requires to anticipate all contents that might ever occur in a summary. Obviously, for completely free meetings this is beyond current state of the art. Thus we restrict ourselves to the AMI scenario and tailor our summary representation model explicitly to the remote control design domain. The task remains challenging though, even in a restricted domain, as the range of topics people might discuss in a meeting is very broad.

We address this challenge with a corpus-driven approach. Building on our successful experiences with ontologies as a knowledge representation formalism in the AMI project, we use the available hand-written summaries in the AMI corpus to derive an ontology for summary contents. We do this in an incremental fashion in which we successively annotate the available summaries and refine our ontology whenever we find it lacking the expressiveness required to represent certain specifics in one of the corpus summaries.

For this task, we reused NXT-based [Carletta et al., 2004] components from previously developed tools together with newly implemented task-specific components to create a new annotation tool "COnAn"[8] (see Fig. 10). To date, we have annotated four full summaries with our ontology-based model.

Our approach bares several advantages. For one, we're obtaining a richly annotated summary corpus as part of the AMI corpus virtually as a by-product of designing the summary representation model. This not only gives us a gold-standard to compare against the output of automatic recognizers of abstract-worthy meeting parts, it also provides training material for any such components that utilize machine learning. Furthermore, it allows the verification of the created model by practically applying it to real data. Shortcomings of the model can be identified immediately and the process can continue with a modified version of the model.

Another benefit is the fact that the hand-annotated summaries can stand in as input for an NLG system, even while the development of components which recognize summary contents automatically from an ongoing or recorded meeting is still ongoing. Thus, although principally arranged in a pipeline, the final text generation system can already be developed in parallel to the content extraction component. We are currently evaluating the best option for the text generation task based on already available NLG systems.

8.2 Extractive Summarization

We have also continued the work on extractive meeting summarisation. We have implemented a baseline from previous work from Edinburgh and extended the simple Maximum Marginal Relevance (MMR) algorithm to a beam search version and have achieved some improvements. We have also worked on the issue of evaluation for summarisation of meetings. The widely used ROUGE metric does not have a well-defined upper-bound. Therefore, we worked on computing the upper-bound for ROUGE score given human

8. Computer-aided Ontology Annotation

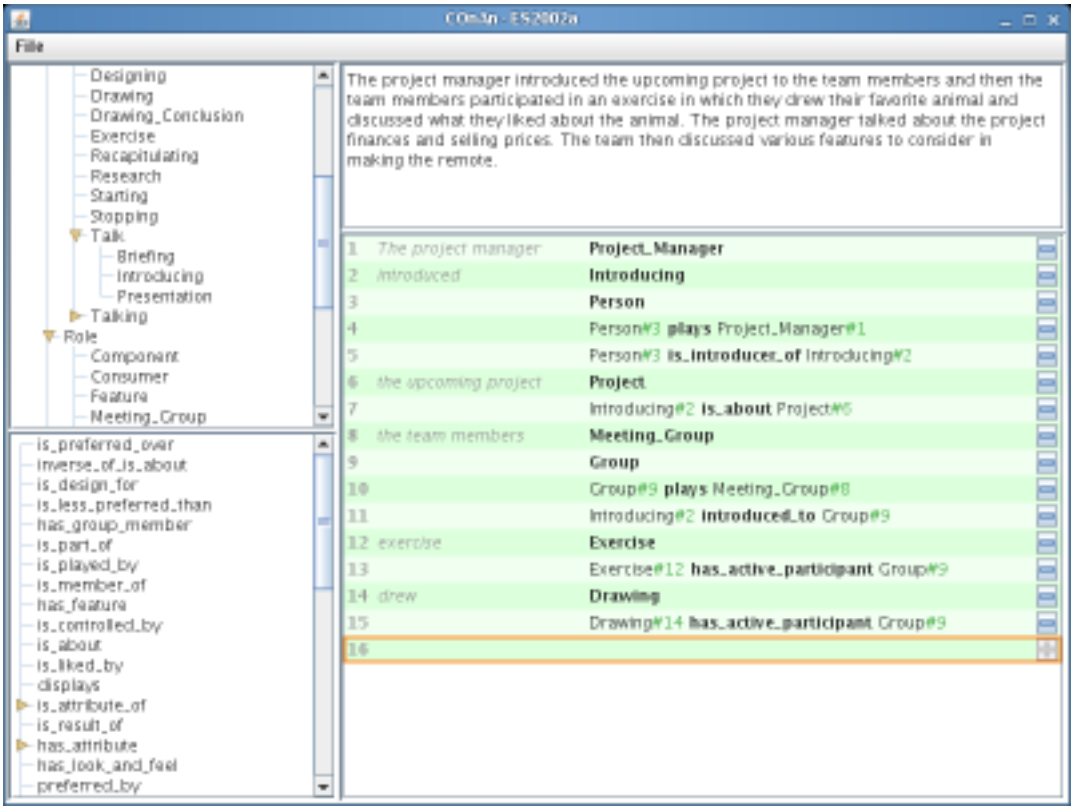

Figure 10: The COnAn annotation tool. *Annotators select text from the summary in the upper right corner and annotate the meaning of the selected text with entities from the ontology in left half of the screen. The list of all annotations is displayed in green.*

summaries, forming what we call the maxROUGE summary. Similarly the weighted precision metric can be maximized (maxWP). We also defined two baselines for extractive summarisation: random selection, and longest sentence selection. We compared all the previous work with these new upper and lower bounds. This work was presented at Interspeech08 Riedhammer et al. [2008a].

We are also working on extracting keyphrases from meetings to use as the seed for improving MMR summarisation and incorporating user feedback into summarisation Riedhammer et al. [2008b]. We have proposed a simple keyphrase extraction algorithm that limits the impact of disfluencies and ASR errors. Keyphrases significantly outperform a simple bag-of-word centroid when used with MMR which show the necessity for query focus in meeting summarization. Therefore, we have designed a prototype user interface to allow exploration and manual refinement of the automatically extracted keyphrases used as a query for summarization. In related work, we have worked on speaker role detection using lexical and social network analysis based features.

Data	Type	Len	Rand	Base	maxR	maxWP
ICSI	A	350 W	0.04	0.06	0.16	
ICSI	E	12.7% W	0.23	0.34	0.46	
ICSI	E	16% W	0.28	0.40	0.55	
ICSI	E	4.2% DA	0.08	0.41	0.50	
ICSI	E	10% DA	0.18	0.64	0.80	
ICSI	L	350 W	0.10	0.49		1.42
ICSI	L	700 W	0.10	0.49		1.20
AMI	L	700 W	0.21	0.71		1.75

Table 18: Baselines and maximums for the AMI and ICSI corpus at according to the type of reference (A=abstract, E=extract, L=links) and the target length (W=words, DA=dialog acts) used in the literature. The systems are: random selection (Rand), longest sentences (Base), maximum ROUGE (maxR) and maximum weighted precision (maxWP).

9 Automatic Video Editing

In this section we describe the online automatic video editing system which has been developed in the second year of AMIDA and some improvements which have been made for the offline system. Video editing can be used in two scenarios: in video conferences and for meeting browsing. Therefore two different system make sense, because for the decision which camera should be shown, different features can be used in each scenario. Thus, the offline version can also use semantic information which can not be extracted online, which can improve the video output.

9.1 Online automatic video editing

We have implemented an interface for the video editing application to connect to the Hub. The data in the Hub are stored as timed-triples. The triple contains entries for object, attribute and value, and is accompanied by a timestamp. But the application for automatic video editing (VideoEditor) uses an XML file as the source, which contains relevant information about the input data, annotated and detected events. A communication protocol was defined to interface the Hub with this application. It defines the format of separate tags stored in the triples in the Hub and it handles the transformation of data read from the XML file into the correspoding timed triples and the following communication with the Hub by means of requests. These requests are processed by an XML parser, which can construct the request and parse the correspoding response to the relevant data entries. This parser can also handle invalid XML, which for example does not contain the termination tags or does not meet the defined DTD.

A comparison of the XML structure with the modified communication protocol for the Hub is in the following listing. [MID] denotes a unique identifier of the meeting, [N] is represented by all indices from 0 to Count of the corresponding section and [...] denotes all different parameters of the respective section. The output data marking the places for the shot boundary are stored with identical names, only instead of [MID] they use [MID-out].

```
<?xml version="1.0"?>
<AVEvents>
 <EventGroups>
  <Group>
   <ID>0</ID><Name>individual_positions</Name>
   <Meaning>State</Meaning><Enabled>1</Enabled>
  </Group>
  <Group>
   <ID>1</ID><Name>Speaking</Name>
   <Meaning>State</Meaning><Enabled>1</Enabled>
  </Group>
 </EventGroups>
 <EventTypes>
  <Type>
   <ID>0</ID><Name>off_camera</Name><Offset>0</Offset>
   <Parameters individual="PM"/>
   <Secondary>
    <Key></Key><Offset>0</Offset><Parameters individual="ID"/>
   </Secondary>
   ...
  </Type>
```

```
  ...
  <Type>
   <ID>8</ID><Name>Start speaking</Name><Offset>0</Offset>
   <Group>1</Group><GroupIndex>0</GroupIndex>
  </Type>
 </EventTypes>
 <File>
  <Source Camera="3">video\ES2003a.Corner_orig.avi</Source>
  ...
  <Source>audio\ES2003a.Mix-Headset.wav</Source>
  <TimeFormat>Milliseconds</TimeFormat>
  ...
  <Event>
   <ID>8</ID><Time>1110415</Time>
   <Text>Yeah? Okay.</Text>
   <Parameters Person="Closeup4"/>
  </Event>
  ...
 </File>
</AVEvents>
::: Example structure of XML input file for video editing application.

object                          attribute
[MID].EventGroups.Group         Count
[MID].EventGroups.Group.[N]     [...]
[MID].EventTypes.Type           Count
[MID].EventTypes.Type.[N]       [...]
[MID].EventTypes.Type.[N]       Parameters.Count
[MID].EventTypes.Type.[N]       Parameters.[N].[...]
[MID].EventTypes.Type.[N]       SecondaryKeys.Count
[MID].EventTypes.Type.[N]       SecondaryKeys.[N].[...]
[MID].EventTypes.Type.[N]       SecondaryKeys.[N].Parameters.Count
[MID].EventTypes.Type.[N]       SecondaryKeys.[N].Parameter.[...]
[MID].Sources.Source            Count
[MID].Sources                   TimeFormat
[MID].Sources.Source.[N]        [...]
[MID].Events.Event              Count
[MID].Events.Event.[N]          [...]
[MID].Events.Event.[N]          Parameters.Count
[MID].Events.Event.[N]          Parameter.[N].[...]
::: Rules for saving XML events into Hub timed-triples.
```

The actions of the current automatic video editing process can be now described by the following pseudo-code:

```
 1 get all appropriate data from hub
 2 if received Data
 3   parse Data into TimedTriples
 4 else
 5   end application
 6 for each InternalStructure find RelevantData
 7   if RelevantData in TimedTriples
 8     fill InternalStructure with RelevantData
 9   else
10     process next InternalStructure
11 run video editing process
```

After the editing process is finished, the resulting set of events–marking the shot boundaries–

Table 19: Evaluation of different model configurations. For the evaluation a combination of global motion and acoustic features were used. In the table MS stands for multi-stream, S describes the number of states per class and M is for the number of Gaussian mixture.

Model configuration	RR (in %)	FER (in %)	AER (in %)
Linear, S=3, M=1	36.9	63.1	10.1
Linear, S=3, M=2	42.3	58.7	9.6
Linear, S=5, M=2	46.6	53.3	15.5
Linear, S=5, M=3	26.4	73.6	8.6
Linear, S=10, M=2	41.6	58.4	21.8
Left-right, S=5, M=2	49.3	50.7	10.1
Ergodic, S=5, M=2	52.3	47.7	12.7
MS linear, S=3, M=2	47.0	53.1	9.6
MS left-right, S=3, M=2	34.5	65.5	18.6
MS ergodic, S=3, M=2	20.5	79.5	25.5
Couple linear, S=3, M=2	31.1	68.9	11.8

is sent back to the Hub to be used for later editing or for comparison/evaluation with other means of editing.

Of course, there still is the possibility of using an XML file instead of Hub, in the case of loading data from XML and save into the Hub and vice versa.

9.2 Offline automatic video editing

We introduce a new approach for feature extraction in Arsić et al. [2007] and we extracted these features for the video editing subset. The new features are based on Global Motions Zobl et al. [2003]. An additional semantic feature is about when a slide change occurs. Therefore the region of the projector screen is analyzed and fast changes are detected. This helps to segment the meeting in a way which is helpful for automatic video editing.

9.2.1 Integrated Segmentation and Classification with Graphical Models

Finding a sequence of phones is enough for speech recognition but in the case of video editing the boundaries of shots are very important for the impression of the video. Therefore we need graphical models (GM) which perform an automatic segmentation and classification at the same time. In Bilmes and Bartels [2005] a basic concept for segmentation is presented. The concept is for speech recognition and so we adapt it for the video editing. We developed several structures which are called linear, left-right and ergodic. Further we extended the structures to a multi-stream and coupled structure.

9.2.2 Evaluation

The results in table 19 are taken from an evaluation where different types of models and model parameters are tested. The first column in the table shows the results which are

achieved for the classification task only. The best model for it is the ergodic one with five states and two Gaussian mixtures with a recognition rate of 52.9%. There is a drop once the models are getting more complex. Therefore the recognition is highly depending on the number of available training data.

The results for the combined task of segmentation and classification are shown in column two and three in table 19. The results show that the ergodic model with five states and two Gaussian mixtures achieves the best frame error rate with 47.4%, therefore it is the best model for video editing. Both models, linear (53.3%) and left-right (50.7%), are worse than the ergodic one, this means that more degrees of freedom helps to select the correct video mode. The low action error rates of most of the models point out that the hardest task is to find the correct boundaries of the segments.

Only the multi-stream linear model achieves slightly better results than the equivalent linear one. The use of multi-stream and coupled models, which are more complex, do not lead to an improvement of the results. The problem with these models is that not enough training data is available and that the training tools are not optimized for high dimensional feature spaces.

9.3 Conclusion and Further plans

The approach with more complex graphical models seems to be successful, but with the small data set what we are currently using the trainings data is not enough. The frame error rate is equal to the best Hidden Markov Model, but the graphical model is using a less dimensional feature space. This leads to the assumption that the graphical models will achieve better rates once the toolkit is more optimized for high dimensional features spaces.

For the online system we will develop fast routines for the feature extraction so that the video editing system gets the required input data. The offline system will be further improved in the way that the slides which are captured during the meeting are added to the output video. Also the performance of the offline system should be increased by using new features and other graphical models.

10 Future Work

In the final year of the AMIDA project, we intend to continue in improving the results in the most important fields, we intend to work on real-time and on-line versions of our systems with minimized latencies, and finally, there are two new lines of research that will be pursued.

Demonstration systems will be enhanced, e.g., a video player will be added to the D^3 demo, however the biggest impact will be seen by the inclusion of further WP5 modules into the prototypes build in WP6. This will extend to remote settings, with one or more participants at different locations and even include mobile scenarios with WP5 components integrated into the mobile meeting assistant prototype.

In an extension of interaction analysis, new work in the final year will cover two fields: the detection of addressees and participation and also an extended profiling of meeting participants.

References

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. International Conference on Machine Learning (ICML)*, pages 282–289, June 2001.

M. Collins. Discriminative reranking for natural language parsing. In *Proc. International Conf. on Machine Learning*, pages 175–182, 2000.

T. Koo and M. Collins. Hidden-variable models for discriminative reranking. In *Proc. Human Language Technology and Empirical Methods in Natural Language Processing*, pages 507–514, October 2005.

L. Shen, A. Sarkar, and F. Och. Discriminative reranking for machine translation. In *Proc. HLT–NAACL*, pages 177–184, May 2004.

A. Dielmann and S. Renals. DBN based joint dialogue act recognition of multiparty meetings. In *Proc. IEEE ICASSP*, volume 4, pages 133–136, April 2007.

A. Dielmann and S. Renals. Recognition of dialogue acts in multiparty meetings using a switching DBN. *IEEE Transactions on Audio, Speech, and Language Processing*, 16 (7):1303–1314, September 2008.

S. Bhagat, H. Carvey, and E. Shriberg. Automatically generated prosodic cues to lexically ambiguous dialog acts in multiparty meetings. In *Proc. International Congress of Phonetic Sciences*, pages 2961–2964, August 2003.

P. Hsueh and J. Moore. Automatic decision detection in meeting speech. In *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–07)*, pages 168–179. Springer, 2007a.

P. Hsueh and J. Moore. What decisions have you made: Automatic decision detection in conversational speech. In *Proc. NACCL–HLT*, pages 25–32, Rochester, NY, USA, April 2007b.

Harm op den Akker and Christian Schulz. Exploring features and classifiers for dialogue act segmentation. In *Proceedings of MLMI*. Springer, 2008.

I.H. witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2 ed.* Morgan Kaufmann, San Francisco, 2005.

Jana Besser. A Corpus-Based Approach to the Classifiation and Correction of Disfluencies in Spontaneous Speech. Bachelor's thesis, University of Saarland / DFKI GmbH, Saarland University, Saarland, Germany, 2006.

Fernanda Ferreira, Ellen F. Lau, and Karl G. D. Bailey. Disfluencies, Language Comprehension, and Tree Adjoining Grammars. In *Cognitive Science*, volume 28, pages 721–749, 2004.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edition, June 2005. ISBN 0-12-088407-0.

Elizabeth Shriberg, Rebecca Bates, and Andreas Stolcke. A Prosody-Only Decision-Tree Model for Disfluency Detection. In *Proc. Eurospeech '97*, pages 2383–2386, Rhodes, Greece, 1997.

R. Grishman B. Favre, D. Hillard, D. Hakkani-Tur H. Ji, and M. Ostendorf. Punctuating speech for information extraction. In *Proceedings of IEEE ICASSP 2008*, Las Vegas, Nevada, USA, April 2008.

B. Favre, D. Hakkani-Tur, S. Petrov, and D. Klein. Efficient sentence segmentation using

syntactic features. In *to appear in 2008 IEEE Workshop on Spoken Language Technology*, 2008.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus. In *Proceedings of the Measuring Behavior Symposium on "Annotating and Measuring Meeting Behavior"*, 2005.

T. Wilson. Annotating subjective content in meetings. In *Proceedings of LREC*, 2008.

R. E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

B. Wrede and E. Shriberg. Spotting "hot spots" in meetings: Human judgments and prosodic cues. In *Proceedings of EUROSPEECH*, 2003.

R. Banse and K. R. Scherer. Acoustic profiles in vocal emotion expression. *Journal of Personality and Social Psychology*, pages 614–636, 1996.

S. Raaijmakers, K. Troung, and T. Wilson. Multimodal subjectivity analysis of multiparty conversation. In *Proceedings of EMNLP*, 2008.

P. Schwarz, P. Matejka, and J. Cernocky. Towards lower error rates in phoneme recognition. In *Proceedings of Intl. Conf. on Text, Speech and Dialogue*, 2004.

T. Wilson and S. Raaijmakers. Comparing word, character, and phoneme n-grams for subjective utterance recognition. In *Proceedings of Interspeech*, 2008.

Jeroen Dral, Dirk Heylen, and Rieks op den Akker. Detecting uncertainty in spoken dialogues. In Kurshid Ahmad, editor, *Proceedings of the Sentiment Analysis workshop at LREC*, pages 72–78, 2008.

Pei-Yun Hsueh and Johanna D. Moore. What decisions have you made? Automatic decision detection in meeting conversations. In *Proc. of NAACL-HLT*, 2007c.

Pei-Yun Hsueh and Johanna D. Moore. Improving meeting summarization by focusing on user needs: A task-oriented evaluation. In *IUI*, In submission.

J. F. Dovidio and S. L. Ellyson. Decoding visual dominance: Attributions of power based on relative percentages of looking while speaking and looking while listening. *Social Psychology Quarterly*, 45(2):106–113, June 1982.

I. Szoke, M. Fapso, M. Karafiat, L. Burget, F. Grezl, P. Schwarz, O. Glembek, P. Matejka, S. Kontar, and J. Cernocky. But system for nist std 2006 - english. In *Proc. NIST SPoken Term Detection Evaluation workshop (STD 2006)*, page 26. National Institute of Standards and Technology, 2006. URL `http://www.fit.vutbr.cz/research/view_pub.php?id=8239`.

I. Szoke, L. Burget, J. Cernocky, and M. Fapso. Sub-word modeling of out of vocabulary words in spoken term detection. In *submitted to 2008 IEEE Workshop on Spoken Language Technology*, page 4, 2008.

J. Carletta, D. McKelvie, A. Isard, A. Mengel, M. Klein, and M.B. Møller. A Generic Approach to Software Support for Linguistic Annotation using XML. In G. Sampson and D. McCarthy, editor, *Corpus Linguistics: Readings in a Widening Discipline*. Continuum International, London and NY, 2004. ISBN: 0826460135.

K. Riedhammer, D. Gillick, B. Favre, and D. Hakkani-Tur. Packing the meeting summarization knapsack. In *Proceedings of Interspeech 2008*, Brisbane, Australia, September 2008a.

K. Riedhammer, B. Favre, and D. Hakkani-Tur. A keyphrase based approach to interactive

meeting summarization. In *to appear in 2008 IEEE Workshop on Spoken Language Technology*, 2008b.

D. Arsić, B. Schuller, and G. Rigoll. Suspicious behavior detection in public transport by fusion of low-level video descriptors. In *Proceedings of the 8th International Conference on Multimedia and Expo (ICME)*, 2007.

M. Zobl, F. Wallhoff, and G. Rigoll. Action recognition in meeting scenarios using global motion features. In J. Ferryman, editor, *Proceedings of the 4th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*, pages 32–36, 2003.

J. Bilmes and C. Bartels. Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, 22(5):89 – 100, 2005.