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 D5.5: WP5 Work in Year 3

Due date: 30/09/2009 **Submission date:** 30/09/2009
Project start date: 1/10/2006 **Duration:** 39 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)	

D5.5: WP5 Work in Year 3

Abstract: This deliverable presents a concise description of the progress in multimodal analysis and structuring made in the final year 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 and AMIDA D5.4. 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 and AMIDA D5.4.

Research results reported include: dialog act classification, disfluencies, subjectivity and sentiment recognition, non-verbal behaviour analysis, a range of work on summarization and video editing. The individual advances are briefly summarized in the next section.

The major overall trends in the final year have been a few new research questions, the improvement of established methods, including the move to remote scenarios, 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 implementation of prototypes in the context of WP6.

1.1 Overview of Results

The work on Dialog Act classification investigates a multi-step classification framework that is motivated by a characteristic set of classifier confusions. While some of these confusions are known to be hard even for humans, the resulting systems shows that previous results can still be improved.

While the AMI(DA) corpus shows a high degree of speech disfluencies (around 14%), other data still contains 5-10% disfluent words. For many algorithms that work on the speech transcript, a cleaned-up version is beneficial and our work on speech disfluency removal addresses this need. The hybrid approach developed for AMIDA, is now combined with a training phase that arranges the modules in an optimal sequence and improves previous results by 35%.

The final year has seen improvements in subjectivity and sentiment recognition. First, a two-step approach has been developed that trains single-feature classifiers and then uses various methods of combining the results. Second, we have worked on detecting agreement and disagreement which is an important aspect of sentiment recognition. Our approach also detects the targets of (dis-)agreement and has won the best student paper award at the ICMI-MLMI 2009 conference. Finally, as a new development, we have used participant subjectivity and involvement for discourse segmentation.

This deliverable reports extensively on our work on non-verbal behaviour analysis. This work investigates the differences in group dynamics between the AMIDA remote meetings and the AMI meetings that were conducted in one room. The results show first the usefulness of nonverbal cues that are computed from speech activity. Also, we found that remote participants talk less and remote meetings need more turns, thus motivating meeting support tools such as AMIDA's user engagement and floor control system.

Work on summarization has been continued and conducted in a number of settings. As always, we distinguish between extractive and abstractive summarization. Two new lines of research use argument diagramming and participant profiling for summarization purposes. The meeting profiler is a combination of browser and extractive summarization

technologies and shows a new type of tag clouds that present the evolvement of key terms over time.

Another approach that also opens the door to include other languages than English has been submitted to VideoCLEF'09 and links ASR results from Dutch television to English Wikipedia pages.

Work on video editing has been improved significantly, combining low level and semantic feature for camera selection.

2 Dialog Acts

This section reports on our efforts to enhance dialogue act (DA) classification. Using the last year's reported state-of-the-art classifier for dialogue acts - the maximum entropy classifier from the Stanford NLP group - we were interested in reducing common inter-class confusions. Looking at the 10 most-common confusions, listed in table 1, we can see that some of them, e.g., (`ass,inf`) or (`sug,inf`) are known to be hard to distinguish by humans (see op den Akker). While we do not expect to reduce the number of confusions for these cases in a statistical significant degree, our hypothesis is that a multi-step classification approach, where special two-class classifier, trained for the disentangling of inter-class confusions would improve the systems overall classification performance.

classes	# of confusions
(ass,inf)	842
(sug,inf)	688
(ass,bck)	636
(inf,fra)	332
(inf,el.inf)	289
(stl,fra)	230
(ass,fra)	180
(ass,stl)	175
(ass,und)	146
(sug,ass)	144
⋮	⋮

Table 1: 10 most-common inter-class confusions, made by the classifier

2.1 Multi-step Classification

In our experiments, we came up with three different configurations for the multi-step classification approach, which we explain here. In each case, we use the state-of-the-art classifier as the main classifier (`MainCL`) and add several *sub-classifier* - one for each of the 10 most common confusions that we consider.

multi1 In version 1 of the multi-step approach, we investigate a setting, where the `MainCL` returns one class *X* out of the whole 15 DA classes that are defined in the AMI scheme. Furthermore, there exist 10 sub-classifier, each trained on the disambiguation of two classes. The current segment gets passed to all sub-classifier that are trained on the class. A majority voting over all results of the sub-classifier is then returned. In a case of a tie, the result of the main classifier is returned.

multi2 Another approach is to classify the current segment with each sub-classifier. This returns 10 values which get then added to the feature set of the main classifier.

multi3 In the last setting, the segment gets passed to the main classifier first. This classifier returns a probability vector over all classes. The two classes with the highest probabilities are taken (e.g., *X* and *Y*) and fed into the corresponding sub-classifier, trained on these two classes. As an additional feature, the feature set of the sub-

classifier is extended with the probability vector, including all 15 values. Finally, the result of this classifier is returned.

2.2 Experimental Results

Unfortunately, we were not able to train the systems on the whole training set, as the necessary wrapping of the internal data structure leads to a memory leak in the system. Hence, we only trained the system on a third of the original corpus for the training process.[1] As a baseline system, we use the maximum entropy classifier from the Stanford NLP group that has been presented as being state-of-the-art in dialogue classification in former deliverables. Table 2 shows the results of that system in the row *mesf*. We compare common machine learning measurements like accuracy and weighted means of precision and recall over all classes, as well as Cohen's kappa value. Furthermore, we present the performance of the particular multi-step classification approaches.

Classifier	Accuracy	avg. Precision	avg. Recall	κ
mesf	58.9 %	57.5 %	58.6 %	0.479
multi1	59.0 %	57.6 %	58.7 %	0.478
multi2	59.0 %	57.6 %	58.7 %	0.478
multi3	**59.9** %	**58.5** %	**59.7** %	**0.493**

Table 2: Experimental results

We can see that the two first versions of the multi-step approaches produce only slightly better results on the evaluation set, which have no statistical significance. However, the last version (`multi3`), outperformed the state-of-the-art system significantly with more than 1% in accuracy and an inevitably higher kappa-value. This supports our hypothesis that not just using the knowledge of the returned class of the main classifier, but also adding its probabilities improves the performance of the corresponding sub-classifier and results in an increased performance.

2.3 Conclusion and Outlook

We have seen that it is possible to outperform the results of a single classification step by using a multi-step approach, where knowledge of common inter-class confusions is used to train specialised classifier to enhance the results. In our setting, the last approach, namely `multi3` produced significantly better results. As these results are based on a first experiment, we want to concentrate our upcoming research to extend this classification approach to find an optimal number of sub-classifier. Furthermore, we want to be able to train the system on the whole amount of training set, which means that we have to come up with a new implementation of the internal data structure.

1. Though the evaluation was done on the whole evaluation set!

3 Disfluencies

Spontaneously spoken language is frequently sprinkled with speech production errors. These errors are also referred to as speech disfluencies and while former research (see Shriberg [1994]) found that 5% - 10% of spoken language are disfluent, a previous analysis of our AMI corpus (see Germesin et al. [2008]) reports an even higher amount of erroneous speech. Transferring this to the level of dialog act segments, about 40.5% of all segments contain at least one disfluency - nearly every second segment. This phenomenon effects many natural language processing systems, as they are usually developed on grammatical correct data and experience a sharp decrease in their performance (see, e.g., Jorgensen [2007]) when faced with ungrammatical, spontaneous speech. To address this problem, we developed a system for the automatic detection and correction of speech disfluencies and presented the last year's deliverable.

Our main interest was to develop a robust system that leverages the heterogeneity of the different disfluency types using different sub-modules, each specialized for its corresponding type. Having a library of modules opens the question how to internally arrange these in the detection system to maximize the detection performance of the system, which was chosen manually in the first version of the system. This forms a disadvantage as the system could not be adapted easily to, e.g., be trained on another language which might introduce other correlations of disfluencies and hence would need new arrangements of the detection modules. That's why we extended our system to learn the arrangement of the modules during the training process, resulting in a system which is able to outperform its old results in both, detection performance and speed.

Meanwhile, we were able to extend the available annotations by annotating data from the AMIDA corpus and results on both data sets will be presented in section 3.4.

3.1 Data

The scheme of the disuency types this study is based on has been developed earlier in the AMI project (see Besser [2006]) and contains a fine-grained annotation scheme of 15 different disuency types (see table 3). Where the original data set contained 28 meetings (22 for the training process and 6 for the evaluation), we now have 45 meetings annotated (33 for the training and the remaining 12 for evaluation purposes), including 5 meetings taken from the AMIDA corpus. That means, we have almost 40,000 dialog acts annotated, respectively 24 hours of speech material.

3.2 Hybrid Disfluency Detection

A thorough investigation of our corpus and the disuency scheme used showed a heterogeneity with respect to how the different disuencies can be detected. This led us to the following design: Easily detectable disuencies should be identied by a simple rule-based approach while the remaining disuencies need a more sophisticated machine learning approach. Furthermore, the usage of different detection techniques, each specialized and fine-tuned on its own disuency domain, yields the advantage of an improved performance in conjunction with a reduced computational overhead at the same time.

class	abbrev.	occur.	example
Hesitation	hesit	9676	This **uh** is an example.
Stuttering	stutter	1147	This is an **exa** example.
Slip Of the Tongue	sot	2112	This is an **y** example.
Discourse Marker	dm	3925	**Well,** this is an example.
Explicit Ed. Term	eet	300	This is **uh** this is an ex.
Disruption	disrupt	3374	This is an example **and I**
Deletion	delete	17	**This really is** this is an ex.
Insertion	insert	240	**This an** this is an example.
Repetition	repeat	4656	**This is** this is an example.
Replacement	replace	801	**This was** this is an example.
Restart	restart	1129	**We should,** this is an example.
Mistake	mistake	1627	This **be** an example.
Order	order	150	This **an is** example.
Omission	omiss	1768	This is **[ ]** example.
Other	other	181	

Table 3: Overview of all Disfluencies of the AMI scheme

Looking at the data, it was a pretty easy task to come up with rules that described disfluencies like, e.g., *Hesitation*, *Stutterings*, *Repetitions* and *Slip-of-the-tongues* very precisely. Finding lexical expression for the detection of the remaining disfluencies is a hard task as their structure also appears often in fluent speech. Nevertheless, we included these rules in the detection part and leave it up to the training process to select the right approach for the detection of each type.

Using the freely available WEKA toolkit for machine learning, we are able to easily test several classifier in a batch process. Besides *lexical* features, we also use *prosodic* and *speaker-related* features to support the classification process.

3.2.1 Hybrid Combination

Besides a detection where all modules work in parallel and detect disfluencies in conjunction, we decided to process the transcribed speech sequential, where each module gets isolated access to the current segment and after that, the text gets passed to the next module in the system's order. In this design, we allow a module to occur more than once in the system, making it possible to include new evidence (already detected disfluencies) in a later detection step. This design leads to the two questions, how the modules should be arranged in the system and, furthermore, which classification approach should be taken for each module. Both issues will be addressed in section 3.3, explaining the concept of **self-arranging modules**. Before that, we give an overview of the different detection modules that have been developed.

3.2.2 Detection Modules

In total, we developed five different detection modules, each responsible for a subset of the disfluency types. We will shortly explain each module and the used features.

SHS This module is responsible for the detection of *Hesitations*, *Stutterings* and *Slip-of-the-tongues* and only uses lexical-based features.

REP Disfluencies of type *Repetition* are detected by this module, using regular expressions, based on lexical input as features.

DNE *Discourse Marker* and *Explicit Editing Terms* are classified by this module. Features using lexical input as well as pause-based features are used here.

DEL Disfluencies of type *Deletion* are detected by this module. There, we use lexical as well as prosodic features.

REV This module has been developed for the detection of four disfluency types, namely: *Insertions*, *Replacements*, *Restarts* and *Other*. Lexical, prosodic as well as speaker-related features are used here.

So far, we do not have modules for the detection of the remaining disfluencies. Types like *Disruption*, *Mistake*, *Order* and *Omission* cannot be detected.

3.3 Self-arranging modules

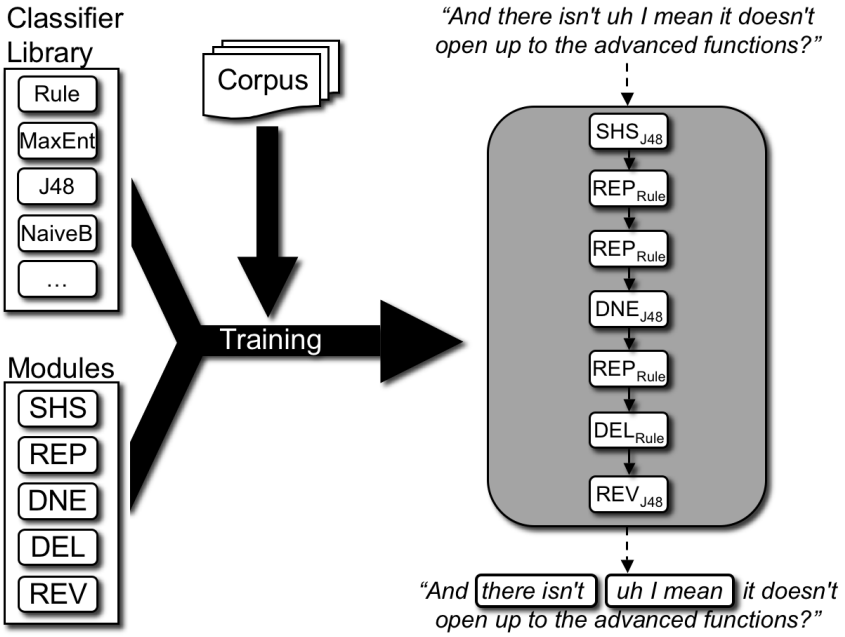

Figure 1: Sketched training process of the hybrid system

In the latest version of the disfluency detection system, we extended the training process in the way that the system has to learn, which internal sequence of (module ⊗ classifier)-combination it has to build. As shown in figure 1, we feed the training process with three resources:

1. a library of different classifiers
2. the list of detection modules
3. the training part of the corpus.

The library of classifiers contains our implementation of the rule-based detection system, the maximum entropy classifier from the Stanford NLP group,[2] the Naïve Bayes classifier and the WEKA implementation of the C4.5 decision tree, namely the J48. Thereby, we

2. http://nlp.stanford.edu/

include different configurations of each classifier, resulting in a library size of more than 4,000 classifier. In general, the number of modules in the resulting system is unknown and depends on the training procedure as well.

3.3.1 Trade-off: Time vs. Truth

The described search space (module ⊗ classifier ⊗ placeInSystem) can be very large and the training process would take an almost infinite amount of time. That's why we had to trade-off the amount of training time that we think is realistic against the need of finding the optimal arrangement of the modules in the system.

As a first step, we exchanged the brute-force approach of finding the best module-arrangement against a greedy hill-climbing selection algorithm that incrementally picks the best module for each place in the system. Algorithm 1, describes this procedure, where each combination of classifier and module gets trained and evaluated on the given data and the best pair is placed in the current place in the system. This step is repeated until no system can be found that could improve the output of the system. During this process, the information of the already found disfluencies gets passed to the upcoming step, to ensure that the module in the next step is trained on the new data. As we are interested in a higher precision that recall, we introduced the α-value[3] that gives more weight to the errors a detection algorithm makes and hence, tweaks the system towards precision.

Algorithm 1 Greedy hill-climbing process of self-arranging modules

```
system ← empty List
repeat
  bestPerformance ← 0.0
  bestModule ← null
  for all Module m : modules do
    for all Classifier c : classifier do
      train(m_c)
      evaluate(m_c)
      performance_α(m_c) ← correct_(m_c) − α ∗ errors_(m_c)
      if performance_α(m_c) > bestPerformance then
        bestPerformance ← performance_α(m_c)
        bestModule ← (m_c)
      end if
    end for
  end for
  if bestModule ≠ null then
    system.add(bestModule)
  end if
until bestModule = null
return system
```

Furthermore, we reduces the set of classifier to a randomly chosen subset for each step in the selection approach. Our experiments have shown that taking 10% of the whole set

3. In our experiments, we used an α-value of 1.5.

of classifier for the training, the maximal performance drop is 2.3% depending on the module.

3.4 Experimental Results

Table 4 compares the baseline of the system (which is a trivial system that would pick the *fluent* class for all words, against the performance of the old system and the results of the new one, including the self-arranging modules. Only the latter one has been trained on the extended data set that includes the AMIDA data. Comparing the results of the old system directly to the new one, trained and evaluated on the old training and evaluation set, it can easily be seen that the new system outperforms the results of the old one by more than 35% (relative improvement). This is mainly because of improvements on the feature set and though we cannot measure the effect of the self-arrangement on the performance but think that this also improved the system's performance as the system is now able to adapt to the data in a better way.

System	Train. data	Eval. data	Acc.	avg. F1	RT-factor
baseline	—	AMI	90.3 %	85.7 %	0.00
		AMI +AMIDA	88.6 %	83.3 %	
old	AMI	AMI	92.9 %	90.5 %	0.42
new	AMI	AMI	95.3 %	94.8 %	0.11
	AMI +AMIDA	AMI	95.1 %	94.7 %	
		AMI +AMIDA	94.5 %	93.5 %	

Table 4: Detection Results

Furthermore, we can see that the new system is faster than the old one. There are mainly two reasons for that: The first reason is that the internal structure of the system has been cleaned up and features gets calculated faster from the data. Secondly, the old system used the part-of-speech (POS) tagging system from the Stanford NLP group. This has been changed in the new version and we now use the tagger from Phan [2006] that uses conditional random fields and is better in terms of execution time without performance loss.

3.4.1 Different Module - Arrangements

Table 5 shows the different configurations of the new version of our system that were trained on the old as well as on the new corpus. There, we can see that both systems share the same modules and classifiers at the same positions but differ in many positions. This may occur because of different classifier that were randomly chosen during the training process, but more likely is because of the bigger amount of training data, which gives the classifier more material to adapt to.

3.5 Future work

So far, we do not have modules for the detection of all disfluencies. Types like *Disruption*, *Mistake*, *Order* and *Omission* cannot be detected. This is definitely a field where research

	AMI		AMI +AMIDA	
Nr.	**Module**	**Classifier**	**Module**	**Classifier**
1.	SHS	J48 '-U -M 2'	SHS	J48 '-L -U -M 3 -A'
2.	REP	RuleMatcher "	REP	RuleMatcher "
3.	DNE	MaxEntStanford "	DNE	MaxEntStanford "
4.	REP	RuleMatcher "	REP	RuleMatcher "
5.	DNE	J48 '-L -S -C 0.45 -M 2 -A'	DNE	J48 '-U -M 5'
6.	SHS	J48 '-U -B -M 4'	SHS	J48 '-L -U -M 4'
7.	REP	RuleMatcher "	REP	RuleMatcher "
8.	DNE	J48 '-U -M 3'	DEL	RuleMatcher "
9.	DNE	MaxEntStanford "	DNE	J48 '-S -R -N 9 -Q 1 -M 2'
10.	DEL	RuleMatcher "	DNE	J48 '-L -U -B -M 3'
11.	SHS	J48 '-U -M 2 -A'	REP	RuleMatcher "
12.	DNE	J48 '-U -M 2'	SHS	J48 '-L -U -B -M 2 -A'
13.	REP	RuleMatcher "	DNE	J48 '-U -B -M 2'
14.	DNE	J48 '-L -U -M 3'	REP	RuleMatcher "
15.	SHS	MaxEntStanford "	DNE	MaxEntStanford "

Table 5: Greedy hill-climbing result of self-arranging modules (first 15 systems)

has to be done, especially once we consider errors introduced by an ASR system that make the sentence ungrammatical. Besides that, driven by further publications from Liu et al. [2005] we hypothesize that integrating other detection approaches like CRFs or HMMs improve the system's performance on the complex disfluencies. However, the modular design of the system allows an easily incorporation of such modules once they are implemented.

4 Subjectivity and Sentiment Recognition

Subjective content includes the opinions, sentiments, agreements, disagreements, and other internal mental and emotional state information that participants express during discussions in a meeting. Previously, we developed an annotation scheme for marking subjective content in meetings and annotated 20 meetings from the AMI corpus [Wilson, 2008]. These annotations were later revised to separate out categories for agreement and disagreement. To date, subjectivity annotations in the AMI corpus have been used in experiments for developing systems to recognise subjective utterances and for distinguishing between utterances expressing positive and negative subjectivity [Wilson and Raaijmakers, 2008, Raaijmakers et al., 2008]. We have also worked on detecting uncertainty in multi-party conversations [Dral et al., 2008].

In the sections below, we present our most recent work on recognizing subjective content in meetings. Specifically, we give our newest results for combining word and sub-word features for recognizing subjective utterances and distinguishing between utterances expressing positive and negative sentiments. We also present new work on identifying agreements and disagreements in meetings. Finally, we give an overview of our recent work that uses features representing participant subjectivity to recognise discourse segments based on speaker intention.

4.1 Subjectivity and Sentiment Recognition

In previous work, we compared word and sub-word features (*n*-grams of words, characters, and phonemes) for recognising subjective utterances and distinguishing between utterances expressing positive and negative subjectivity [Wilson and Raaijmakers, 2008, Raaijmakers et al., 2008]. This work showed that sub-word features indeed were useful for subjectivity and sentiment recognition in multi-party conversation, but it was the combination of word, character, and phoneme *n*-grams that achieved the best results.[4] However, there are questions that remain. What is the best way of combining these low-level features? Will the usefulness of the features bear out when ASR transcription and automatic phonemes are used?

4.1.1 Experiments and Results

We used the 13 meetings from previous work for our experiments, performing 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. For the experiments summarised below, we report results using manual dialogue act segments as the unit of classification. We have also experimented using spurts as the unit of classification with similar results.

The experiments involve two steps. First, a classifier is trained for each type of feature (word *n*-grams, character *n*-grams, or phoneme *n*-grams) using BoosTexter's [Schapire and Singer, 2000] AdaBoost.MH. These are referred to as the single-source classifiers.

4. Acoustic-prosodic features were also evaluated but were found to be less useful.

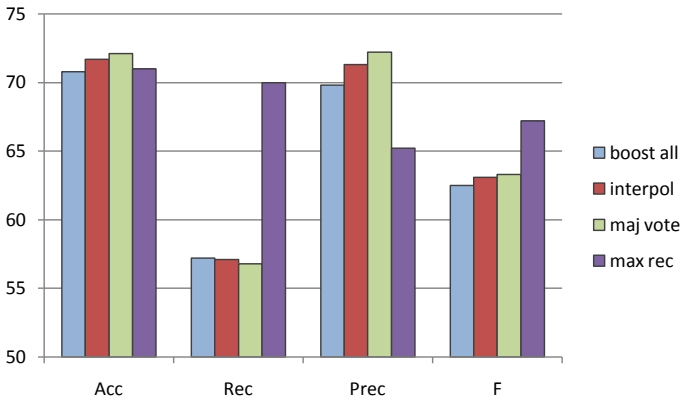

Figure 2: Subjective vs. Non-subjective: reference transcripts

Then, we compare the performance of various methods of aggregating the model predictions.

boost all – With this method, BoosTexter is given all the features together to produce a single classifier. This is the baseline.

linear interpolation – This is the method originally used in Raaijmakers et al. [2008]. The models produced when training the single-source classifiers are combined using a simple linear interpolation strategy. In the present binary class setting, BoosTexter produces two decision values, one for each class. For every individual single-source classifier, 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.

majority vote – With this method of aggregation, the outputs of the three single-source classifiers are all considered, and the class that is predicted by the majority is taken as the final prediction.

maximiise recall – This method of aggregation maximises the recall. Essentially, if any of the three single-source classifiers predicts that an utterance is subjective (or negative for the positive-negative classification task), then the final prediction is subjective (or negative).

Figures 2 and 3 show the results for classifying subjective sentences. Although performance is slightly lower when n-gram features are extracted from the ASR, the trends in performance are similar. Interestingly, the classifiers with the simpler aggregation methods, majority vote and maximise recall, do better than the more complex linear-interpolation method. The method that maximises recall does indeed produce large improvements in recall. Although the improvements in recall come at the expense of precision, the drop in precision is much lower than the gains in recall.

Figures 4 and 5 show the results for distinguishing between positive and negative sub-

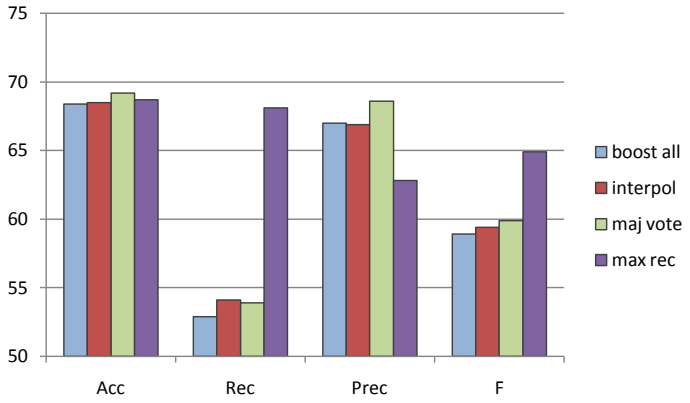

Figure 3: Subjective vs. Non-subjective: ASR transcripts

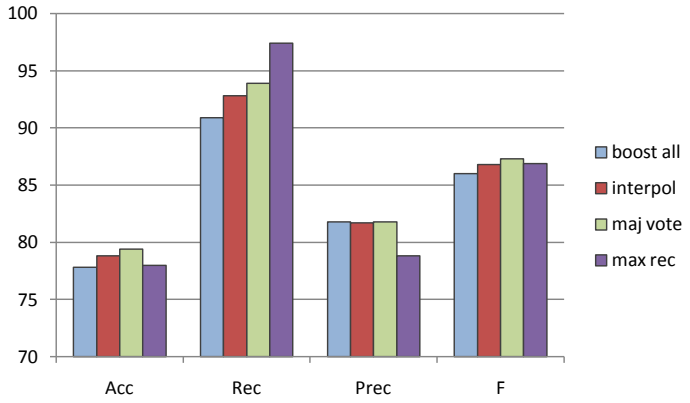

Figure 4: Positive vs. Negative Subjective: reference transcripts

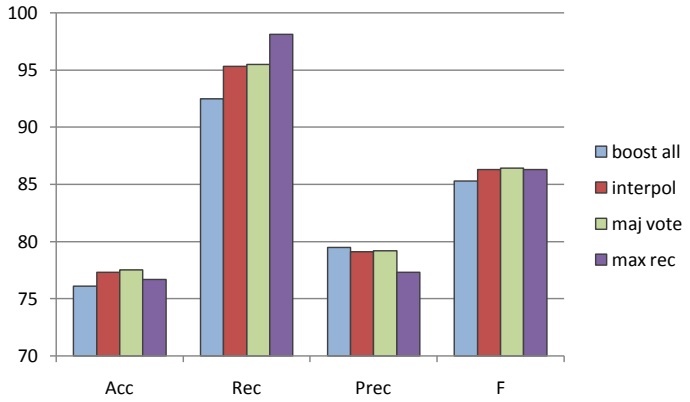

Figure 5: Positive vs. Negative Subjective: ASR transcripts

jectivity. As before, the method for maximizing recall does yield good improvements in recall. However unlike the subjectivity classification experiments, the improvements in recall are nearly offset by the decrease in precision. Overall, there is less difference between the various aggregation methods for positive-negative classification than subjectivity classification. Nonetheless, there are clear gains to be made by aggregating the results of the various classifiers as opposed to giving all the features to the learning algorithm to produce a single classifier.

4.2 Agreement and Disagreement Detection

One key type of subjective content in meetings are agreements and disagreements. To date, there has been only a small amount of work on agreement[5] detection in meetings, and no work on detecting the target of agreements (what the agreement is about). For this work, we used agreement annotations in the AMI corpus to develop automatic systems for these tasks, using a combination of high-precision rules and machine learning classifers. The automatic systems exploit a wide variety of features, including lexical, prosodic, and structural features.

In the sections below, we give an overview of our experiments and results for detecting agreements and the speaker targets of agreements. Full details of the research can be found in Germesin and Wilson [2009].

4.2.1 Data

These experiments use the 20 meetings from the AMI corpus annotated with subjectivity information [Wilson, 2008]. The agreement and disagreement annotations and their targets are part of the second version of the AMIDA subjectivity annotations. Within the 20 meetings, there are a total of 636 agremeents and 70 disagreements. This is a very skewed class distribution, and may reflect the way the corpus was built. The participants never met before the recording of the meeting. This may make them hesitant to disagree with each other in an effort to be polite.

For our later experiments, we randomly divided the data into two sets: 80% for training and 20% evaluation. Table 6 shows the division of the meetings between the two sets. The training data was used for the development of rules and features for the automatic experiments.

train	ES2002a-c ES2008b-d ES2009a-d IS1003a-b IS1003d TS3005a TS3005c-d
test	ES2002d ES2008a IS1003c TS3005b

Table 6: Corpus distribution in training and evaluation part

The dialog that is presented below is an excerpt from the AMI corpus showing an example of an annotated agreement. The text in bold is the actual agreement of speaker B, whereas the *target* of the agreement, what B is agreeing with, is underlined. We refer to the speaker of the target utterance as the *target speaker*.

5. We will use the term agreement for both agreements and disagreements unless a distinction is necessary.

... ...
A: Finding them is really a pain.
D: Hm.
A: I mean, when you want it, it's kicked under the table or so.
B: **Yeah, that's right**.
... ...

4.2.2 Features for Learning

Lexical Lexical features are features that incorporate information about the spoken words. In this work, we use words from the manual transcription, using the POS tagger from the Stanford NLP group[6] as presented in [Toutanova and Manning, 2000a] to obtain part-of-speech tags. We derive features such as the number of (content) words in a segment and the first, second and last word of a segment. We also use various keywords for agreements, as well as positive and negative polarity words.

To calculate keywords, we follow Hillard et al. [2003], who chose keywords based on an 'effectiveness ratio,' which is the frequency of an *n*-gram in a given class divided by its frequency in all other classes combined. Table 7 shows the top keywords according to their effectiveness ratio for both agreement and disagreements. Unfortunately, even the top keywords for the disagreement class had very low effectiveness ratios. Thus, only agreement keywords were used in our experiments.

agree		disagree	
6.0	think so too	0.43	⟨s⟩ no no
2.5	yep yep	0.43	no no
2.5	that's right	0.41	no no no
2.5	definitely ⟨/s⟩	0.10	no
2.0	that's true	0.09	⟨s⟩ no

Table 7: Keywords for agreement/disagreement.

The positive and negative polarity words are taken from the MPQA subjectivity lexicon [Wilson et al., 2005].

Prosodic Prosodic features describe information about timing like the duration of a segment or pauses and the speech rate of the speaker. We also use data about the pitch and energy of the voice. In a pre-processing step, the raw prosody data is normalized using z-normalization ($z = \frac{x-\mu}{\sigma}$). In addition to standard features like the minimum, maximum and mean values, we also calculated the kurtosis and skewness of the values, all for the first word of the segment and for the whole segment.

Dialogue Act Labels We hypothesized that contextual information like the labels of the current and surrounding dialog acts could be an important source of information for

6. http://nlp.stanford.edu/

recognising agreements. We use the manually annotated dialog act labels from the AMI corpus in our experiments which contain a total of 15 types of labels.[7] Table 8 gives the distribution of agreements and disagreements for each dialog act label, followed by the total number of segments with that label in the training data. We can see that about 69% (= 373/549) of all agreements and 61% (= 40/66) of all disagreements are labeled as assessments. Interestingly, more than 13% of all agreements were (manually) tagged as backchannels - short and unsubstantial segments. This reflects the ambiguity that sometimes exists for short utterances between what is an agreement and what is only a backchannel, for example with the very frequent word *yeah*. However, overall, when considering the total number of backchannels, we see the actual amount of confusion is small.

DA label	agree	disagree	total
assessment	373	40	2996
backchannel	72	1	1460
inform	39	9	4280
suggestion	20	2	1322
fragment	19	5	1256
understanding	10	1	475
⟨*all other (9)*⟩	37	8	3359
total	549	66	15148

Table 8: Distribution of DA labels for agreements and disagreements.

Table 9 shows the distribution of the top seven dialog act labels for the targets of both agreements and disagreements. There, we can see that the majority (more than 77%) of the targeted segments are either giving information (inform), giving an assessment, or making a suggestion. We use this information in the last part of our system, where we detect whom the current speaker is actually agreeing with.

DA label	target	
	count	[%]
inform	211	32.50
assessment	159	24.50
suggestion	131	20.20
fragment	40	6.16
elicit assessment	39	6.01
stall	24	3.70
elicit inform	21	3.24
⟨*all other (8)*⟩	23	3.69

Table 9: Distribution of DA labels for targets.

Structural With structural features, we refer to features that take the context of the current segment into account. Thereby, we compare local features that are part of the previously described feature types to the ones from the surrounding segments. It is important

7. Guidelines for Dialogue Act V1.0, Oct 13, 2005. `http://mmm.idiap.ch/private/ami/annotation/dialogue acts manual 1.0.pdf`

to model these structural features speaker-dependent, that means, the information about a (possible) speaker change has to be included to model the speakers' interactivity.

4.2.3 Automatic Recognition

Figure 6 sketches the architecture of the system we developed for the detection of agreements. As the figure shows, there are two main parts to the system. The first part, which we call *agreement detection* involves two steps. First, we use a set of high-precision rules to label all segments as *not (dis)agreement*, *agreement*, or *unclassified*. This information is then fed into a second classification step, which uses supervised machine learning for the final detection of agreements and disagreements. In the last part we perform the target detection, in which we determine who the agreement or disagreement actually was directed towards.

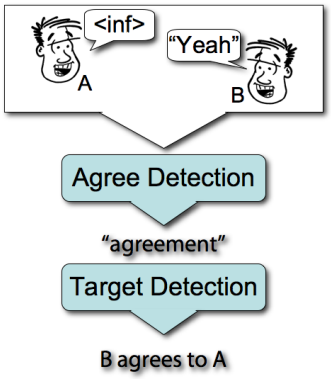

Figure 6: Sketched design of the detection system.

High-Precision Rules (HPR) The first step of agreement detection uses a set of high-precision rules to provide an initial labeling of the data. When examining the training data, we found that quite often there are places within the meeting were agreements are rarely found, e.g., if only one person has been talking, subsequent utterances by this same person are not likely to be agreements. The class labels provided by this step, *not (dis)agreement*, *agreement*, and *unclassified* are incorporated in the subsequent step as features.

The high-precision rules are described below. When classifying the segments, the rules are applied in a cascading manner in the order listed. If a segment s is tagged as *unclassified* by a given rule, the next rule will then try to classify it.

1. **No-Target:** If all preceeding segments (window of 13 segments) that are longer than 6 words also have the same speaker as s, then $tag(s) = not(dis)agreement$, else $tag(s) = unclassified$.
2. **DA-Label (agreement):** If s is an *elicit, offer,* or *be-negative* dialog act, then $tag(s) = not(dis)agreement$, else $tag(s) = unclassified$.
3. **DA-Label (target):** If the previous 4 segments do not contain *comment-about-understanding, be-positive, be-negative, elicit-suggestion, offer, backchannel, other,* or *elicit-understanding* dialog acts, then $tag(s) =$ *not (dis)agreement*, else $tag(s) = unclassified$.

4. **Silence:** If there was a pause of more than 15 seconds before s, then $tag(s) = not(dis)agreement$, else $tag(s) = unclassified$.
5. **Length:** If length of s is greater than 15 words, then $tag(s) = not(dis)agreement$, else $tag(s) = unclassified$.
6. **Subjectivity:** If s does not contain any subjective content (based on manual annotations), then $tag(s) = not(dis)agreement$, else $tag(s) = unclassified$.
7. **Agreement:** If a special agreement n-gram, e.g., 'i agree', 'i think so', occurs within s then, $tag(s) = agreement$, else $tag(s) = unclassified$.

Machine Learning For the second step of our detection system, we trained and evaluated classifiers using two different supervised machine learning approaches: Decision Trees (DT) and Conditional Random Fields (CRF). In this step, each dialog act segment is classified as an *agreement*, *disagreement*, or *other*.

We used lexical features like the occurrence of special n-grams and the number of repeated words compared to previous segments, as well as prosodic features, namely duration- and pause-based features. We also included the labels of the dialog act segments and the output of the HPR classifier. By its very structure, the CRF is able to model inter-dependency between features. However, the DT is not able to do this. Therefore, for the DT classifier we created special features to capture the more complex inter-dependencies, using a window of ten segments around the current segment being classified.

Target Detection In addition to the automatic detection of agreements, it is also important to know who the *target speaker* of the agreement is. The target speaker of an agreement is represented by an index of speakers counting backwards from the current segment. Specifically, we defined the current speaker as having index '0,' the next (other) speaker as index '1,' and so forth. To help illustrate this, we show below the speaker indexing for dialog act segments from the example in section 4.2.1.

> ...
> *(speaker index 3)* A: Finding them is really a pain.
> *(speaker index 2)* D: Hm.
> *(speaker index 1)* A: I mean, when you want it, it's kicked under the table or so.
> *(speaker index 0)* B: **Yeah, that's right**.
> ...

Table 10 shows the distribution of target speaker indexes for agreements and disagreements in our corpus. A baseline approach for target speaker detection would just be to use the last speaker as the target. We hypothesize, that using adjacency pair information will improve this baseline and our algorithm for identifying target speakers is given below.

4.2.4 Evaluation

High-Precision Rules Table 11 shows the performance of the HPR classifier. Recall that the rules are applied in a cascading manner. For example, rule two is only applied to the segments that remain unclassified by rule one.

speaker index	agree [%]	disagree [%]
1	66.0	44.9
2	26.8	44.9
3	7.2	10.2

Table 10: Addressee distribution

Algorithm 2 Pseudocode of Target Detection

```
for Segment s in meeting do
  if s is (dis)agree then
    {Use AP if available}
    for last 10 segments p do
      if (s,p) isAP then
        s.addressee = getIndex(p.speaker)
      end if
    end for
    {Fallback}
    if s has no addressee yet then
      s.addressee = getIndex(getLastSpeaker())
    end if
  end if
end for
```

There are 3,920 segments in the test data. After applying the HPR classifier, 3,362 segments are classified as having no agreement/disagrements, 4 as agreements, and 554 remain unclassified.

(Dis-)Agreement Detection The performance of the (dis-)agreement detection for each approach is given in Table 12. The baseline is the most frequent class. Given the highly skewed nature of the data, it is not sufficient to report only accuracy. Thus, we also report precision, recall and F-measure (F1) as well as kappa.

If we compare the results of both approaches, it is interesting to see that we observe a drop in precision if we use the prior knowledge of the HPRs and in fact, the best system was the CRF that was built without the HPRs. Looking at the recall, we can see that the usage of the HPRs did actually increase the performance of the system but unfortunately

number	name	correct	wrong
1	No-Target	740	12
2	DA-Label (src)	295	2
3	DA-Label (tar)	274	2
4	Silence	1	0
5	Length	141	5
6	Subjectivity	1890	0
7	Agreement	4	0

Table 11: Evaluation of High-Precision Rules

	Baseline	Conditional Random Field		Decision Tree	
		w. HPRs	wo. HPRs	w. HPRs	wo. HPRs
Accuracy [%]	97.8	98.0	98.1	97.8	97.8
Prec. (agree) [%]	0.0	57.6	58.8	45.0	48.5
Rec. (agree) [%]	0.0	36.3	34.6	31.1	42.4
F1 (agree) [%]	0.0	44.5	43.5	36.8	45.2
κ	0.00	0.40	0.39	0.35	0.40
RT Factor	0.000	0.005	0.005	0.010	0.030

Table 12: Segment-based evaluation of (Dis-)Agreement Detection

introduced more false positives. The DT that was trained without the HPRs performed best considering the F1 score but again with the cost of a high number of false positives. To sum up, we do not have one best-performing system. Instead, we have two systems with different strengths: The CRF classifier with higher precision and the DT classifier with the higher recall.

Unfortunately, neither of the systems were able to detect any disagreements in the evaluation set (though they performed very well on the training data). This is most probably due to the fact that there were just not enough examples to train on. We think that the detection would perform better if we split the system into two systems, each responsible for its own type of agreement/disagreement.

Target Detection Table 13 shows the results of the target speaker detection. We can see from the results that our approach significantly outperformed the baseline of 64.5% with more than 80.3% accuracy and a kappa value of 0.52. Although these results rely on manual adjacency-pair annotations, they are a very promising indicator that automatic adjacency pairs will also prove quite useful for this task.

		classified as						
		1	2	3	Base [%]	Ac. [%]	F1 [%]	κ
real	1	163	0	1	64.5	80.3	86.9	0.52
	2	38	40	0			67.2	
	3	10	1	1			14.2	

Table 13: Evaluation of Target Detection

4.3 Discourse Segmentation Using Participant Subjectivity and Involvement

Speaker subjectivity also plays a role when analyzing conversations in terms of communicative activities. Examples of communicative activities include relating a personal experience, making a group decision, committing to future action, and giving instructions. These kinds of events are part of participants' common-sense notion of the goals and accomplishments of a dialogue.

Activities like these commonly occur as cohesive *episodes* of multiple turns within a conversation [Korolija, 1998]. They represent an intermediate level of dialogue structure

– greater than a single speech act but still small enough to have a potentially well-defined singular purpose. They have a temporal granularity of anywhere from a few seconds to several minutes. Ultimately, it would be useful to use descriptions of such activities in automatic summarization technologies for conversational genres. This would provide an activity-oriented summary describing what 'happened' that would complement one based on information content or what the conversation was 'about'.

The work described below investigates the usefulness of features that represent **participant subjectivity** and **participant involvement** for the intentional segmentation of dialogue. Participant subjectivity concerns attitudinal and perspectival relationships *towards* the dialogue content. This includes properties such as whether the utterance expresses the private mental state of the speaker, or the participants' temporal relationship to a described event. Participant involvement concerns the roles participants play *within* the dialogue content, e.g., as the agent of a described event. We refer to linguistic features that express participant subjectivity and participant involvement as **participant-relational features**.

We view the conversational activities in which we are interested as representing a coarse level of the intentional structure of the dialogue [Grosz and Sidner, 1986]. Thus, developing a system for intentional segmentation of dialogue is a first step towards recognising the types of conversational activities that are our focus.

In the sections below, we give an overview of our experiments and results for using participant-relational features for intentional discourse segmentation. Full details of the research can be found in Niekrasz and Moore [2009].

4.3.1 Data

The data we use for our experiments is a corpus of 20 conversational monologues (i.e., one person speaks while another listens, giving backchannels and other non-verbal listening cues) known as the Pear Stories [Chafe, 1980]. Chafe asked subjects to view a silent movie and then summarize it for a second person. Their speech was then manually transcribed and segmented into prosodic phrases. This resulted in a mean 100 phrases per narrative and a mean 6.7 words per phrase. In later work by Passonneau and Litman [1997], each narrative was segmented by seven annotators according to an informal definition of communicative intention. Each prosodic phrase boundary was a possible discourse segment boundary. Using Cochran's Q test, they concluded that an appropriate gold standard could be produced by using the set of boundaries assigned by at least three of the seven annotators. This is the gold standard we use for this work.

While our long term goal is to apply our techniques to multi-party conversations, using this dataset is a stepping-stone towards that end which allows us to compare our results with existing intentional segmentation algorithms.

4.3.2 Segmentation Algorithm

The basic idea behind our algorithm is to distinguish utterances according to the type of activity in which they occur. To do this, we identify a set of utterance properties relating to participant subjectivity and participant involvement, according to which activity types may be distinguished. We then develop a routine for automatically extracting the

linguistic features which indicate such properties. Finally, the dialogue is segmented at locations of high discontinuity in that feature space. The algorithm works in four phases: pre-processing, feature extraction, similarity measurement, and boundary assignment.

Pre-processing For pre-processing, disfluencies are removed by deleting repeated strings of words and incomplete words. The transcript is then parsed, and a collection of typed grammatical dependencies are generated. Finally, a verb tense and aspect are tagged using an automatic system.

Feature extraction Feature extraction is the most important and novel part of our algorithm. Each prosodic phrase is assigned values for five binary features. The extracted features correspond to a set of utterance properties which were identified manually through corpus analysis. The first four relate directly to individual activity types and are therefore mutually exclusive properties.

first-person participation [1P] – helps to distinguish meta-discussion between the speaker and hearer (e.g., "Did I tell you that?")

generic second-person [2P-GEN] – helps to distinguish narration told from the perspective of a generic participant (e.g., "You see a man picking pears")

third-person stative/progressive [3P-STAT] – helps to distinguish narrative activities related to "setting the scene" (e.g., "[There is a man — a man is] picking pears")

third-person event [3P-EVENT] – helps to distinguish event-driven third-person narrative activities (e.g. "The man drops the pears")

past/non-past [PAST] – helps to distinguish narrative activities by temporal orientation (e.g. "The man drops the pears" vs. "The man dropped the pears")

Feature extraction works by identifying the linguistic elements that indicate each utterance property. First, prosodic phrases containing a first- or second-person pronoun in grammatical subject or object relation to any clause are identified (common fillers like *you know*, *I think*, and *I don't know* are ignored). Of the identified phrases, those with first-person pronouns are marked for 1P, while the others are marked for 2P-GEN. For the remaining prosodic phrases, those with a matrix clause are identified. Of those identified, if either its head verb is *be* or *have*, it is tagged by TTT2 as having progressive aspect, or the prosodic phrase contains an existential *there*, then it is marked for 3P-STAT. The others are marked for 3P-EVENT. Finally, if the matrix clause was tagged as past tense, the phrase is marked for PAST. In cases where no participant-relational features are identified (e.g., no matrix clause, no pronouns), the prosodic phrase is assigned the same features as the preceding one, effectively marking a continuation of the current activity type.

Similarity measurement Similarity is calculated according to the cosine similarity $cos(v_i, c_i)$ between the feature vector v_i of each prosodic phrase i and a weighted sum c_i of the feature vectors in the preceding context. The algorithm requires a parameter l to be set for the desired mean segment length. This determines the window $w = floor(l/2)$ of preceding utterances to be used. The weighted sum representing the preceding context is computed as $c_i = \sum_{j=1}^{w}((1 + w - j)/w)v_{i-j}$, which gives increasingly greater weight to more recent phrases.

Boundary assignment In the final step, the algorithm assigns boundaries where the similarity score is lowest, namely prior to prosodic phrases where cos is less than the first $1/l$ quantile for that discourse.

4.3.3 Experiments and Results

We compares the performance of our novel algorithm (which we call NM09) with a naive baseline and a well-known alternative method – P&L's co-reference based NP algorithm. To our knowledge, P&L is the only existing publication describing algorithms designed specifically for intentional segmentation of dialogue. Their NP algorithm exploits annotations of direct and inferred relations between noun phrases in adjacent units.

The NP algorithm requires co-reference annotations as input, so to create a fully-automatic version (NP-AUTO) we have employed a state-of-the-art co-reference resolution system [Poesio and Kabadjov, 2004] to generate the required input. We also include results based on P&L's original human co-reference annotations (NP-HUMAN).

For reference, we include a baseline that randomly assigns boundaries at the same mean frequency as the gold-standard annotations, i.e., a sequence drawn from the Bernoulli distribution with success probability $p = 0.169$ (this probability determines the value of the target segment length parameter l in our own algorithm). As a top-line reference, we calculate the mean of the seven annotators' scores with respect to the three-annotator gold standard.

For evaluation we employ two types of measures. On one hand, we use $P(k)$ [Beeferman et al., 1999] as an error measure designed to accommodate near-miss boundary assignments. It is useful because it estimates the probability that two randomly drawn points will be assigned incorrectly to either the same or different segments. On the other hand, we use Cohen's Kappa (κ) to evaluate the precise placement of boundaries such that each potential boundary site is considered a binary classification. While κ is typically used to evaluate inter-annotator agreement, it is a useful measure of classification accuracy in our experiment for two reasons. First, it accounts for the strong class bias in our data. Second, it allows a direct and intuitive comparison with our inter-annotator top-line reference. We also provide results for the commonly-used IR measures F_1, recall, and precision. These are useful for comparing with previous results in the literature and provide a more widely-understood measure of the accuracy of the results. Precision and recall are also helpful in revealing the effects of any classification bias the algorithms may have.

The results are calculated for 18 of the 20 narratives, as manual feature development involved the use of two randomly selected narratives as development data. The one exception is NP-HUMAN, which is evaluated on the 10 narratives for which there are manual co-reference annotations.

The mean results for the 18 narratives, calculated in comparison to the three-annotator gold standard, are shown in Table 14. NP-HUMAN and NM09 are both superior to the random baseline for all measures ($p \leq 0.05$). NP-AUTO, however, is only superior in terms of recall and F_1 ($p \leq 0.05$).

	$P(k)$	κ	F_1	Rec.	Prec.
Human	.21	.58	.65	.64	.69
NP-HUMAN	.35	.38	.40	.52	.46
NM09	.44	.11	.24	.23	.28
NP-AUTO	.52	.03	.27	.71	.17
Random	.50	.00	.15	.14	.17

Table 14: Mean results for the 18 test narratives.

4.3.4 Discussion

The results indicate that the simple set of features we have chosen can be used for intentional segmentation. While the results are not near human performance, it is encouraging that such a simple set of easily extractable features achieves results that are 19% (κ), 24% ($P(k)$), and 18% (F_1) of human performance, relative to the random baseline.

The other notable result is the very high recall score of NP-AUTO, which helps to produce a respectable F_1 score. However, a low κ reveals that when accounting for class bias, this system is actually not far from the performance of a high recall random classifier.

Error analysis showed that the reason for the problems with NP-AUTO was the lack of reference chains produced by the automatic co-reference system. While the system seems to have performed well for direct co-reference, it did not do well with bridging reference. Inferred relations were an important part of the reference chains produced by P&L, and it is now clear that these play a significant role in the performance of the NP algorithm. Our algorithm is not dependent on this difficult processing problem, which typically requires world knowledge in the form of training on large datasets or the use of large lexical resources.

It is important to place our experiment on intentional segmentation in context with the most commonly studied automatic segmentation task: topic-based segmentation. We conducted an additional experiment comparing the results of our novel algorithm with exiting topic segmentation methods. We employ Choi's implementations of c99 [Choi, 2000] and TEXTTILING [Hearst, 1997] as examples of well-known topic-oriented methods. While we acknowledge that there are newer algorithms which improve upon this work, these were selected for being well studied and easy to apply out-of-the-box. Our method and evaluation is the same as in the previous experiment.

The mean results for the 18 narratives are shown in Table 15, with the human and baseline score reproduced from the previous table. All three automatic algorithms are superior to the random baseline in terms of $P(k)$, κ, and F_1 ($p \leq 0.05$). The only statistically significant difference ($p \leq 0.05$) between the three automatic methods is between NM09 and TEXTTILING in terms of F_1. The observed difference between NM09 and TEXTTILING in terms of κ is only moderately significant ($p \leq 0.08$). The observed differences between between NM09 and c99 are minimally significant ($p \leq 0.24$) .

The comparable performance achieved by our simple perspective-based approach in comparison to the lexical-semantic approaches used by the topic segmentation systems sug-

NP-auto	$P(k)$	κ	F_1	Rec.	Prec.
Human	.21	.58	.65	.64	.69
NM**09**	.44	.11	.24	.24	.28
C99	.44	.08	.22	.20	.24
TEXTTILING	.41	.05	.18	.16	.21
Random	.50	.00	.15	.14	.17

Table 15: Results comparing our method to topic-oriented segmentation methods.

gests two main points. First, it validates our novel approach in practical applied terms. It shows that perspective-oriented features, being simple to extract and applicable to a variety of genres, are potentially very useful for automatic discourse segmentation systems.

Second, the results show that the teasing apart of topic-oriented and intentional structure may be quite difficult. Studies of coherence at the level of short passages or episodes [Korolija, 1998] suggest that coherence is established through a complex interaction of topical, intentional, and other contextual factors. In this experiment, the major portion of the dialogues are oriented towards the basic narrative activity which is the premise of the Pear Stories dataset. This means that there are many times when the activity type does not change at intentional boundaries. At other times, the activity type changes but neither the topic nor the set of referents is significantly changed. The different types of algorithms we have tried (i.e., topical, referential, and perspectival) seem to be operating on somewhat orthogonal bases, though it is difficult to say quantitatively how this relates to the types of "communicative task" transitions occurring at the boundaries. In a sense, we have proposed an algorithm for performing "activity type cohesion" which mimics the methods of lexical cohesion but is based upon a different dimension of the discourse. The results indicate that these are both related to intentional structure.

4.4 Continuing Work

We are continuing to work on improving subjectivity recognition in general, as well as improving the identification of specific types of subjectivity, namely positive and negative sentiments and (dis)agreements. One challenge for these tasks is the very skewed distributions of the categories in the data. Thus, one focus of our immediate work is investigating which methods for dealing with highly skewed datasets are best for these tasks.

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

5 Non-verbal Behaviour Analysis

5.1 Targeted Objectives and Summary of Achievements

The overall goal of our work this year was to continue investigating methods to estimate dominant behaviour and also to work on other tasks, while still concentrating how non-verbal cues can be used. Namely, we collected a new set of annotations on analysing cohesion in the meetings, and also investigated the difference in group behaviour between meetings with one remote participant compared to the co-located scenario. In summary, our work produced the following achievements:

- **Investigation of estimating the dominance using visual focus of attention (VFOA)**. A thorough analysis of how estimates of VFOA can affect measures of visual dominance. The best performing feature and VFOA estimationg method gave a performance of 82% for estimating the most dominant person and 77 % for estimating the least dominant person. This work is described in Section 5.2.3.
- **Investigation of cohesion in teams**. A new set of annotations were collected to examine the level of cohesion in the AMI meetings. This is described in Section 5.3 and showed that the best feature was able to estimate both low and high cohesion meetings with a classification accuracy of 93%, which improved over the random performance of 50%.
- **Investigation of and estimation of co-located vs remote meetings**. Non-verbal cues were used to investigate the differences in behaviour of co-located meeting participants compared to the scenario when one of the team is located remotely. Experiments were carried out with the AMIDA data on three different tasks: (i) distinguishing between remote and co-located meetings, (ii) inferring the remote meeting (given 2 co-located and 1 remote from the same team), and (iii) predicting the remote participant in the remote meeting scenario. For task (i), the best performance of 70% was attained but this was only slightly above chance (66%). Using a more context-dependent information, the performance increased to 81% (baseline 33%). Finally, the remote participant was predicted at best, 50% of the time, which improves on the 25% baseline. The experiments and results are described in Section 5.4.

5.2 Estimating Dominance using estimates of visual focus of attention

In the last year we have expanded initial studies on estimating dominance using estimates of visual focus of attention (VFOA) to consider other gaze-based measures of dominance. We carried out an extensive study in this area by also considering different dominance tasks. In addition, we also considered how different methods of estimating the VFOA might affect the dominance estimation performance.

5.2.1 Summary of Dominance Estimation Tasks

We carried out experiments on both the task of estimating the most and least dominant person from our annotated meeting data. Tests were carried out on all meetings where the annotators were in full agreement on who the most or least dominant person was. In total, 34 meetings were used for estimating the most dominant person and 31 for estimating the least dominant person.

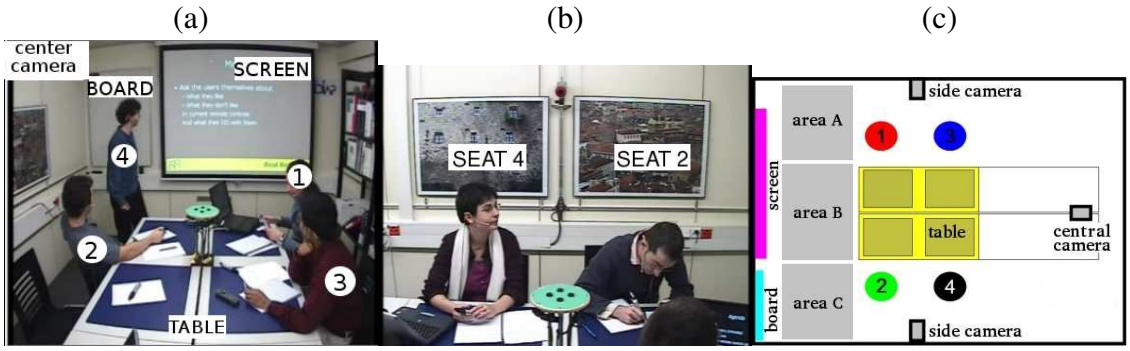

Figure 7: VFOA estimation setup: (a) display an image of the center view camera is used to estimate projection screen activities. (b) displays an image one of the side view camera that are used to track people's head poses. (c) gives a plan of the meeting room.

5.2.2 Visual Focus of Attention Estimation

We extend our recent work (Ba and Odobez [2008a]) to estimate the joint focus state of all participants. We rely on several features. The visual focus of attention (VFOA) of a person is defined the person or the object which is at the center his of visual attention. The visual focus of attention is defined by the eye gaze direction. In the the context of our study, the direct estimation of the gaze direction from the eye features is impossible because of the low resolution imagery conditions. In such a context, people's head poses can be used as a proxy to estimate their focus of attention. In our scenario we have identified a set of 8 visual targets of interest for each one of the four meeting participants. Each participant has as potential visual targets the three other participants, the table, the projection screen, the white board. Whenever a person was not focusing any of the previously mentioned visual targets, he was considered as being unfocused. Fig. 7 gives the meeting room setup and the position of the VFOA targets in the room. We used the framework presented in S. O. Ba et al. [2009] to estimate the joint focus state of all the meeting participants. We followed two approaches, the first one extracts the VFOA solely from people's head pose, the second one uses together with people's head poses meeting contextual information relating gaze and conversation dynamics.

VFOA Modeling from head pose
VFOA from head requires the estimation of people's head poses and information about their location in the room.

People's head poses and location estimation: To estimate peoples head location and pose we rely on a Bayesian formulation of the tracking problem solved through particle filtering techniques (Ba and Odobez [2005]). We applied our tracking method to track people when they are visible in the side cameras (see Fig. 7). At each time t, the tracker outputs the head locations in the image plane and the head poses o_t^k (characterized by a pan and tilt angle) of people visible in the side view cameras.

The location x_t^k of a participant k is a discrete index which takes four values: his seat (seat k) when he is seated, or the center of one the presentation areas A, B or C showed in Fig. 7) when he is standing. This location is estimated by assuming that when people are away from their seats they are standing in one of the area A, B or C. Thus, when they are

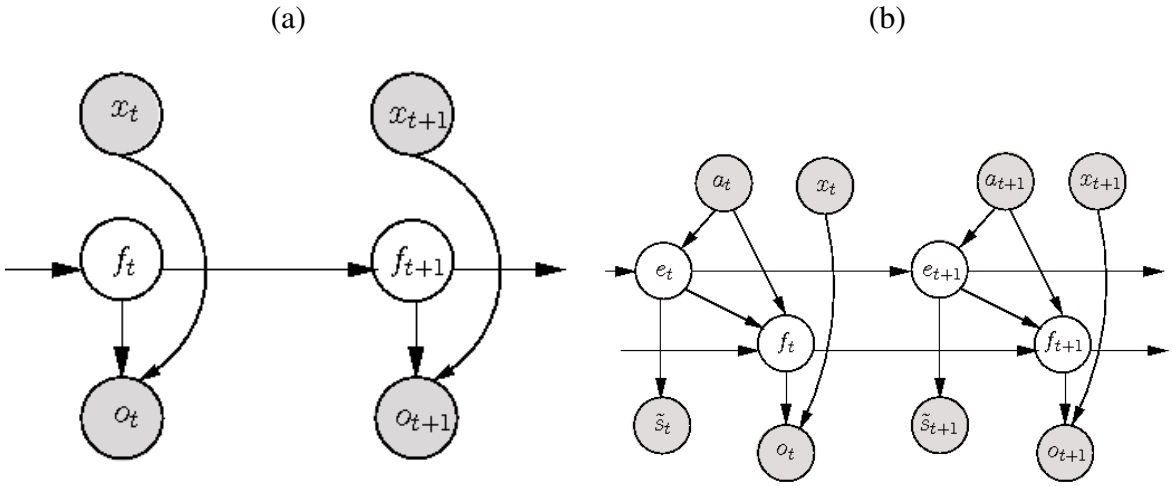

Figure 8: VFOA Recognition graphical model: (a) from head pose, (b) from head pose and meeting contextual information

away from their seats to make presentations, they are localized from the central camera view using motion detection.

The model: The problem of estimating people's VFOA from their head pose can be formalized as follows. If we denote by $f_{1:T}$ the temporal sequence of the four meeting participants' VFOA state. At each time, the joint VFOA state $f_t = (f_t^1, f_t^2, f_t^3, f_t^4)$, group the VFOA states f_t^k of the individuals k. We assume available the head poses $o_{1:T}$ and the locations $x_{1:T}$ of the four meeting participants. The VFOA estimation problem can be stated in a Bayesian framework as finding the hidden state sequence maximizing the posterior probability distribution $p(f_{1:T}|o_{1:T}, x_{1:T})$ which can be factorized as:

$$p(f_o)\prod_{t=1}^{T} p(o_t|f_t, x_t)p(f_t|f_{t-1}) \tag{1}$$

In Fig. 8(a) is depicted the graphical model giving the Markovian independence assumptions corresponding to the factorization in Eq. 1. The posterior probability distribution is defined by three elements.

The initial state distribution $p(f_o)$ initial state is modeled as a uniform distribution on the set of possible visual targets.

The state transition probabilities $p(f_t|f_{t-1})$ models the temporal evolution of the hidden state. We model this transition table in a way that it set higher probability to remain in the same state than to jump to another state. This way of modeling enforces smoothness in the state transitions.

The observation model $p(o_t|f_t, x_t)$ relates people's head pose to their VFOA states given the visual targets location. We modeled the observation model as follows. First we assumed that given their VFOA and the persons' location, people's head poses were independent. This assumption gave the following factorizarion $p(o_t|f_t, x_t) = \prod_{k=1}^{4} p(o_t^k|f_t^k, x_t)$. Then, when person k focuses at another person j, we defined the observation model as a Gaussian distribution:

$$p(o_t^k|f_t^k = j, x_t) = \mathcal{N}(o_t^k; \mu_{k,x_t^j}, \Sigma_k^j) \tag{2}$$

where μ_{k,x_t^j} is the Gaussian mean which models the mean head pose when the person at seat k looks at person j located at position x_t^j, and Σ_k^j is the covariance of the Gaussian distribution. If the visual target j is an object (table, white board, projection), the observation model is defined as a Gaussian distribution $p(o_t^k|f_t^k = j, x_t) = \mathcal{N}(o_t^k; \mu_{k,j}, \Sigma_k^j)$. The probability of being unfocused, $p(o_t^k|f_t^k = \text{unfocused}, x_t)$ is modeled as a uniform distribution.

Model inference: Given a test data sequence, we estimate the optimal VFOA state sequence given the data by first applying an unsupervised hidden Markov model (HMM) maximum a posteriori (MAP) adaptation framework to estimate the best parameters for the head pose observation model (Ba and Odobez [2008b]). Then given the optimal parameters we apply Viterbi search to find the optimal state sequence.

VFOA modeling from head pose and meeting context

The VFOA model presented in Section 5.2.2 make use only of head pose information. Studies about gaze dynamics have shown that VFOA dynamics is strongly dependent on the meeting conversation context: people gaze at person when he is speaking, people gaze at the projection screen when there is new slide. In this Section we described a model that accounts for the effects of the context on people's visual attention.

The model: As in the previous section our goal is to estimate the people's VFOA sequence $f_{1:T}$ given their head poses and locations. We also assume that we are given the sequence of people's speaking proportion $\tilde{s}_{1:T}$ that can be straightforwardly estimated from their speaking statuses. The variable $\tilde{s}_t^k$ denotes the percentage of time person k has been speaking during an interval of time centered at time t. We are also given the sequence of projection screen activity variable $a_{1:T}$ where a_t denotes at time t the time that has passed since the last slide change. a_t is estimated by detecting the slide changes using the center view camera (see Fig. 7). Sharp temporal intensity variations area above a given threshold in the projection screen are considered to be due to slide changes. We introduce conversational context in our VFOA estimation model by jointly estimating together with the sequence of VFOA states, the sequence of meeting conversational events states $e_{1:T}$. A conversational event e_t at time t, denotes the hidden conversation type (silence/-monologue /dialog/discussion) occurring over a time window, that affects the dynamics of the gaze and speech patterns. Thus, our goal is to estimate the optimal sequence of hidden states maximizing the posterior probability distribution $p(f_{1:T}, e_{1:T}|o_{1:T}, x_{1:T}, e_{1:T}, a_{1:T})$ The graphical model in Fig. 8(b) describes the independence assumptions between the model variables which allows the following expansion:

$$p(f_0)\prod_{t=1}^{T} p(o_t|f_t, x_t)p(\tilde{s}_t|e_t)p(f_t, e_t|f_{t-1}, e_{t-1}, a_t) \tag{3}$$

The posterior probability distribution is composed of four terms: $p(f_0)$ the initial probability distribution and the head pose observation model $p(o_t|f_t, x_t)$ have already been defined in Section 5.2.2. The two remaining terms are the speaking proportion observation model $p(\tilde{s}_t|e_t)$ which relates the speaking proportion to the conversational event and $p(f_t, e_t|f_{t-1}, e_{t-1}, a_t)$ the joint VFOA and conversational event transition model allows to include conversational context to the VFOA state transition.

The speaking proportion observation model:. We assume that given the conversational

event, people's speaking proportions are independent. This allows us to factorize the observation model according to the individual as $p(\tilde{s}_t|e_t) = \prod_{t=1}^{4} p(\tilde{s}_t^k|e_t)$. Then we defined the speaking proportion observation model of each individual as a Beta distribution[8]

$$p(\tilde{s}_t^k|e_t = l) = \mathcal{B}(\tilde{s}_t^k, L\eta_{k,l}, L(1-\eta_{k,l})) \tag{4}$$

where $\eta_{k,l}$ is the probability of person k to speak during the conversational event l, and L is an hyper-parameter used to tune the skewness of the model. We adopted Beta distributions because they are well-suited to model proportions.

The joint VFOA and conversational event transitions: models the temporal evolution of the hidden state. It can be written as $p(f_t|f_{t-1}, e_t, a_t)p(e_t|e_{t-1}, a_t)$ to show the dependency of the VFOA dynamics on the conversational events and the projection screen activity, and the dependency of the conversational event dynamics on the projection screen activity.
We defined the conversational event dynamics as

$$p(e_t|e_{t-1}, a_t) = p(e_t|e_{t-1})p(e_t|a_t). \tag{5}$$

The first term models the smooth temporal transition between conversational events. The second term, learned from training data, models the dependency of the conversational event to the projection screen activities.
The VFOA dynamics are defined as

$$p(f_t|f_t, e_t, a_t) = p(f_t|f_{t-1})p(f_t|e_t, a_t). \tag{6}$$

The first term, already presented in Section 5.2.2, models the smooth transitions of the VFOA state. The second term, learned from training data, models the probability of focus states to occur given conversational events and slide activities configurations. This term allows to model the fact that when a person is speaking he has a high probability to be the focus of the listeners, or when there is a new slide projected on the screen people would look at it with a higher probability than any other targets. As illustrated by Fig 9, the tables learned from training data capture very well the relationship between the VFOA, the speaking activities and the projection screen activities.

Model inference: Because of the dimensionality of the hidden state, applying Viterbi search as done for the inference of the first model is computationally very expensive. To reduce the computations, we adopted an iterative two pass inference scheme. Iteratively we will first, using Viterbi search, estimate the optimal conversational sequences given the VFOA states [9]. Then we restimate the VFOA state sequence given the conversational events. During the VFOA estimation, as for the inference of the first model, we apply an HMM MAP estimation scheme to estimate the best head pose observation model parameters given all the observations and the contextual information. Then we use Viterbi search to find the optimal VFOA state sequence.

5.2.3 Estimating Visual Dominance

Received Visual Attention

Dovidio and Ellyson [1982] suggested that someone who receives more visual attention

8. A Beta density distribution is defined as $\mathcal{B}(x, p, q) \propto x^p(1-x)^q$

9. For the initialization of the algorithm we initialize inference scheme by estimating the conversation event sequence only from the speaking proportions.

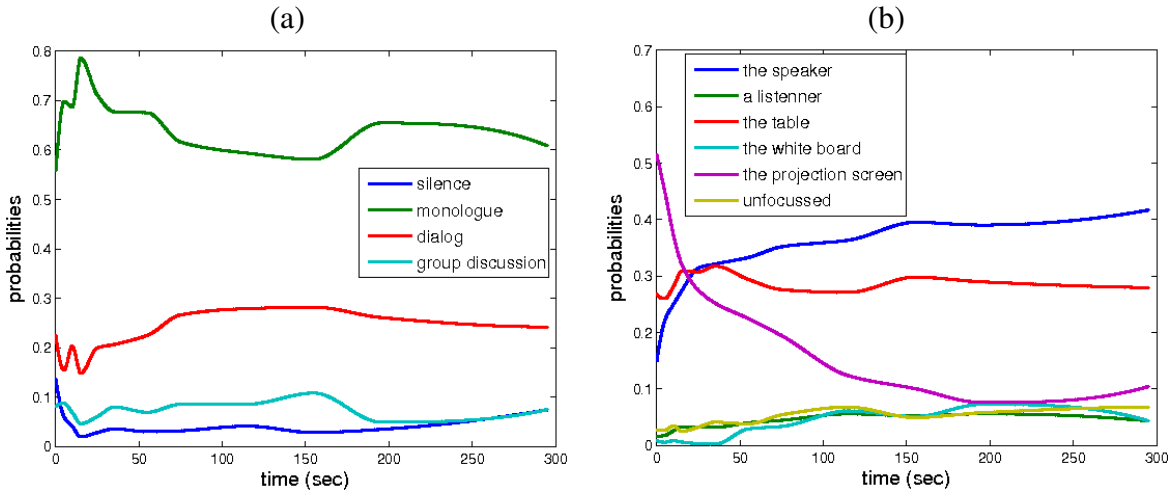

Figure 9: Meeting contextual priors: (a) gives the conversational events probabilities given the time that has passed since the last slide change. (b) gives the probability for a listener during a monologue focusing at the speaker, another listener, the table, the white board, the projection screen or being unfocused given the time since the last slide change.

is perceived to be more dominant. 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

$$\mathbf{RVA}_{\ i} = \sum_{t=1}^{T} \left(\sum_{j=1, j\neq i}^{M_p} \delta(f_t^j - i) \right), \ i = 1, \ldots, M_p \quad , \tag{7}$$

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. Fig. 10 (a) shows two examples of different scenarios for participant A. They show that the VFOA of each participant on A is counted, regardless of whether A is speaking or not. We also encoded the ability of each person to 'grab' visual attention by considering the *RVA* feature in terms of events rather than frames.

From the Dyadic to Multi-Party VDR

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. Thus the dyadic VDR is defined as:

$$VDR_{ij} = \frac{\sum_{t=1}^{T} s_t^i \delta(f_t^i - j)}{\sum_{t=1}^{T} (1 - s_t^i) \delta(f_t^i - j) s_t^j.} \tag{8}$$

It encodes the displayed dominance through either active or passive participation where the first term of the denominator determines if the person is silent, the second term iden-

tifies who they are looking at, and the final term identifies if the person receiving the attention is speaking or not. However, since this expression works only for dyadic conversations. When there are more participants, we can try to combine this measure in different ways so that we have a representation of how visually dominant each participant is to all others.

Combined Pairwise VDR Measures
A simple extension of the VDR would treat each possible pair of participants as single entities such that a person cannot be considered dominant unless they are visually dominant over all participants. We consider an extension so the VDR in terms of a set of measures of dyadic visual dominance before combining the ratio into a multi-party version. The first approach we use assumes that the most dominant person is clearly visually dominant over all other participants. Therefore, we considered the VDR for each pair-wise combination of the 4 participants to see who had the higher VDR between the two and accumulated a set of binary decisions about who was more dominant. The person with the higher VDR was given a vote of 1 and the person with the most votes was considered the most dominant. We refer to this measure later as **MostVotesPairVDR** .

We also consider two measures of multi-party VDR where each pairwise relation is also considered equally. First, we combined them by taking the mean VDR across all pairwise combinations:

$$\mathbf{MeanVDR}_{\ i} = \frac{1}{M_p - 1} \sum_{j=1, j\neq i}^{M_p} VDR_{ij} \tag{9}$$

Then we also considered the product of all the pairwise VDRs.

$$\mathbf{ProductVDR}_{\ i} = \prod_{j=1, j\neq i}^{M_p} VDR_{ij} \tag{10}$$

Intuitively, the first measures how visually dominant the person is on average while the second measures the similarity between the visual dominance between participant i and the others. The second measure draws on observations in social psychology that dominant primates tend to monitor subordinates more evenly while subordinates will look more at the more dominant one.

VDR as a Group Measure
Clearly other definitions of the multi-party VDR are also possible. We extend the VDR to a group scenario (**GVDR**) where each person is considered in relation to all others rather than each other person in turn. 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 as shown in Fig. 10(b). The **GVDR** for person i is:

$$\mathbf{GVDR}_{\ i} = \frac{\mathbf{GVDR^N}_{\ i}}{\mathbf{GVDR^D}_{\ i}} \ , \tag{11}$$

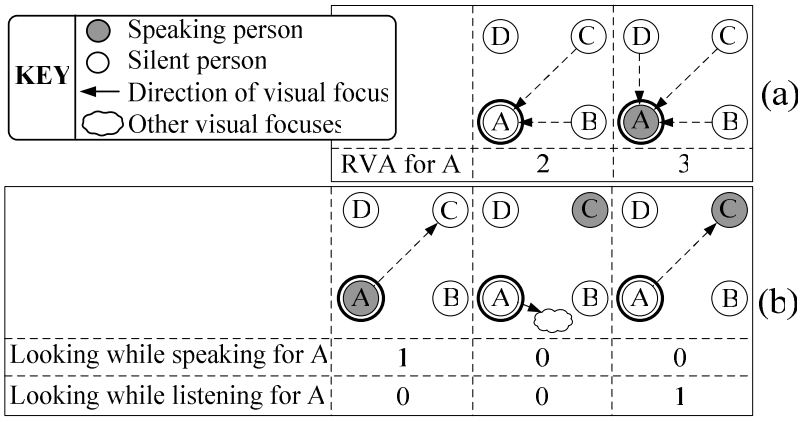

Figure 10: Example scorings of *RVA* and *GVDR* for person A (highlighted node), at time *t*: (a) two examples of *RVA*; (b) three example scenarios for looking-while-speaking and looking-while-listening.

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

$$\mathbf{GVDR^N}_{\ i} = \sum_{t=1}^{T} s_t^i \sum_{j=1, j\neq i}^{M_p} \begin{cases} 1 \text{ if } & f_t^i = j \\ 0 & \text{otherwise} \end{cases} , i = 1 \ldots M_p \tag{12}$$

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

$$\mathbf{GVDR^D}_{\ i} = \sum_{t=1}^{T} (1 - s_t^i) \sum_{j=1, j\neq i}^{M_p} \begin{cases} s_t^j \text{ if } & f_t^i = j \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

There are other ways to quantify visual dominance in terms of the visual dominance ratio. We also considered different ways the numerator and denominator of the GVDR could be combined. Since Dovidio and Ellyson found that dominant people tend to have a high amount of looking while speaking and low amounts of looking while listening, the GVDR was redefined such that the absolute difference was used to measure relative levels of dominance:

$$\Delta\mathbf{GVD} = \mathbf{GVDR^N} - \mathbf{GVDR^D} \tag{14}$$

Earlier work (Hung et al. [2008]) has suggested that trying to detect the listening is difficult since the behaviour is not easy to define. Therefore, we also have a feature which just quantifies the amount of time a person is looking at others while speaking, which is determined by the numerator of the **GVDR** ($\mathbf{GVDR^N}$). This roughly approximates speaking length but puts emphasis on dominant people addressing others rather than just speaking.

5.2.4 Other Measures of Visual Dominance

As well as the VDR, there are other ways of measuring the visual dominance of a person. One method, suggested by Dovidio and Ellyson to be observed in primate behaviour,

concerns the number of times that primates monitor others in the group. Those who monitor more evenly tend to be more dominant. We define this as a form of monitoring while speaking:

$$\textbf{MeanAudienceMonitor}_{\,i} = \frac{1}{|\mathcal{T}_\mathbf{i}|}\sum_{r}^{\mathcal{T}_\mathbf{i}} n_r \tag{15}$$

where $\mathcal{T}_\mathbf{i}$ is the set of all speaker turns of participant i and n_r is the total number of people are observed during turn r. A speaker turn is defined as a continuous period of time for which $s_t^i = 1$.

We can also measure visual dominance by seeing how well the most dominant person can regulate the conversation. It was found by Kalma [1992] that gazing in triads was a powerful signal for how the the floor was distributed amongst the participants in the group. It was suggested that there was a period of prolonged gaze, lasting about 2s at the end of a speaker's turn where they will look at the person they expect to speak next. Using this information, we hypothesised that those who were most successful at passing the floor onto the person they looked at next, could be considered the most dominant person in the conversation. We considered this successful floor yielding to be defined by considering the person the speaker gazed at the longest during 2s after the end of their speaker turn. If the person who spoke next was the person who received a period of prolonged gaze from the speaker, this was considered a successful floor transition. Only speaker turns that were longer than 2s were considered. We will refer to this feature as **SuccessfulFloorYield**.

Interpersonal Influence

We compared our features to those suggested by Otsuka et al. [2006] who estimated the visual focus of attention of 4 participants from a set of conversational regimes. From these features, they defined some measures of interpersonal influence. We use here their measures of incoming and outgoing influence and also the influence balance. Firstly, they define influence as a sum of the influence during dialogues ($I_D(i, j)$) and monologues($I_M(i, j)$).

$$I(i, j) = I_M(i, j) + I_D(i, j) \tag{16}$$

The influence during monologues for person i is then defined as:

$$I_M(i, j) = \frac{1}{T}\sum_{t} \delta(f_t^j - i)\delta(c_t - R_i^m) \tag{17}$$

where c_t is the conversational regime that is estimated at the current time, R_i^m is the label for a i having a monologue, and T is the total time of the meeting that is begin considered. The influence during dialogues between i and j is defined:

$$I_D(i, j) = \frac{1}{T}\sum_{t} \delta(c_t - R_{ij}^d) \tag{18}$$

where the dialogue conversational regime between person i and j is defined by R_{ij}^d. We can then define the incoming influence as:

$$I^{in}\,(i) = \sum_{j, j\neq i}^{M_p} I_(j, i) \tag{19}$$

The outgoing influence is thus defined:

$$I^{out}(i) = \sum_{j,j\neq i}^{M_p} I(i,j) \tag{20}$$

The influence balance is then defined as:

$$\Delta I(i) = I_{out}(i) - I_{in}(i) \tag{21}$$

In our case, since the model for estimating the VFOA is different, we use the estimates of the conversational events e_t to determine regime type. All conversational events with more than 2 people involved were not considered.

5.2.5 From Frame to Event-based Features

As well as treating most of the features on a frame basis, we also considered the same features as events. This is an interesting way to represent the data since it quantifies how often someone is able to cause a change in the group dynamics. This is particularly useful since for some of our data, certain people in the meeting had to give a presentation and could be presenting at the slide screen for long periods of time without interruptions from others. The presenter may not be the most dominant person but could be represented as this since they would, by default, have the most visual attention in terms of time. Event-based features can be considered intuitively to be the ability of each person to grab the visual attention.

To calculate the events, a frame-based representation of each feature is created. Then, an event is determined by:

$$Event_i \equiv g_{i,t} \neq g_{i,t-1} \tag{22}$$

For the VDR-based features, g was considered in turn, to be each pair-wise measure of the numerator or denominator of the term. For the features based on interpersonal influence, the pair-wise I_D and $_M$ events were calculated first and accumulated to form the three features in Equations 19,20, and 21. It should be noted that $g_{i,t} \neq g_{i,t-1}$ for cases where even though the feature type may remain constant (e.g. $\mathbf{GVDR^N}$ remains true) but the circumstances by which the feature is calculated has changed (so there is a change of who a person is speaking to), then this is also considered an event.

5.2.6 Results: Estimating the Most Dominant Person

To estimate the most dominant person, for all the features that were presented, the person with the highest value was taken to be the most dominant. There was one exception for the $\mathbf{GVDR^D}$ feature where the person with the lowest value was taken to be the most dominant person. For some features, the possibility of participants having the same feature value was possible. Therefore, to account for these ties, a score of the reciprocal of the number of ties was used for these meetings.

Estimating visual dominance using manual VFOA annotations
Experiments to compare the different measures of visual dominance were initially carried out using manual annotations of the VFOA. A summary of some of the results are

Type	Method	Active	Passive	Equation
Pairwise VDR	**MostVotesPairVDR**	✓		8
	MeanVDR	✓		9
	ProductVDR	✓		10
Group VDR	**GVDR**	✓		11
	ΔGVD	✓		14
	GVDR$^{\mathbf{N}}$	✓		12
	GVDR$^{\mathbf{D}}$	✓		12
Other	**RVA**		✓	7
	SuccessfulFloorYield	✓		n/a
	MeanAudienceMonitor	✓		15
	I^{in}	✓		19
	I^{out}		✓	20
	ΔI	✓		21

Table 16: Summary of features.

shown in the 'M' columns of Table 17. We compared frame-based results with those that used events. Events are defined as a continuous period of time for which a particular set of properties hold. We used this to make a different representation of a person's level of visual dominance to ensure that those who were considered to be more dominant had a quite active role throughout the meeting where their focus or speaking status was constantly changing.

From the results, we can see that using manually annotated VFOA features, the best performance was obtained for the **GVDR$^{\mathbf{N}}$** feature with 79.4 % classification accuracy. The feature performs the best for both frame-based and event-based versions of the feature. Both I^{in} and **GVDR$^{\mathbf{D}}$** performed quite badly, indicating the difficulty of accurately quantifying when people are listening. For all VDR-based features, the event-based features performed better (except for **GVDR$^{\mathbf{N}}$** where the performance was equally good, and **GVDR$^{\mathbf{D}}$** where using events led to worse performance). It was noted that the performance difference between the frame and event-based features was much bigger between the pairwise VDR features compared to the group-based ones. This indicates an inherent stability in the group-based features compared to the pairwise VDR features. Indeed, the group-based VDR features also performed better overall than the pairwise ones. This could be due to the annotations being considered also on a global basis. Aside from the VDR-based features, both ΔI and I^{out} features also performed quite well.

Estimating the Most Dominant from context independent Estimates of VFOA
A selection of the results are summarised in the 'H' columns of Table 17. The best performance was achieved by **GVDR$^{\mathbf{N}}$** with a performance of 77.9% for the event-based features. Overall observations indicate that the group-based VDR features perform better than pairwise. However, the performance of **RVA** drops considerably, which could be explained by increased noisy estimates from times when the person is not speaking. Again, I^{out} and ΔI perform very well but I^{in} performs badly. Overall, the performance using automatic estimates of the VFOA doest not improve on using manual estimates. Encouragingly, in some cases however the decrease in performance is minor.

MostDominant	Frame			Events		
VFOA Method	M	F	H	M	F	H
MostVotesPairVDR	66.7	76.5	56.9	75.5	76.5	56.9
MeanVDR	58.8	61.8	58.8	67.6	52.9	52.9
ProductVDR	67.6	67.6	58.8	76.5	67.6	55.9
GVDR	73.5	**82.4**	**76.5**	76.5	73.5	61.8
ΔGVD	73.5	79.4	**76.5**	76.5	**77.9**	73.5
GVDR$^{\mathbf{N}}$	**79.4**	70.6	73.5	**79.4**	73.5	**77.9**
GVDR$^{\mathbf{D}}$	41.2	50.0	38.2	36.8	32.4	35.3
RVA	58.8	67.6	17.6	67.6	29.4	11.8
SuccessfulFloorYield	n/a	n/a	n/a	57.4	56.6	64.0
MeanAudienceMonitor	n/a	n/a	n/a	42.2	45.6	35.3
I^{in}	32.4	23.5	17.6	29.4	38.2	17.6
I^{out}	76.5	79.4	73.5	73.5	76.5	76.5
ΔI	76.5	74.3	74.3	75.0	72.8	74.3

Table 17: Summary of results for the estimating the Most Dominant person when the method of estimating the VFOA is varied. M:Manual, F:Full context model, H:Head pose only. The best performance for each VFOA method is shown in bold.

Estimating the Most Dominant from VFOA estimates using contextual cues
Next we conducted the same experiments but used the automatically generated VFOA using contextual cues, as shown in the 'F' columns of Table 17. Here **GVDR** performed the best in the frame-based case with 82.4%. **ΔGVD** performed best for the event-based feature at 77.9%. **GVDR$^{\mathbf{D}}$** again performed worst among all VDR-based features. It is interesting to see that there was no deterioration in performance when using the these automated rather than manual annotations of the VFOA. This could be explained by the contextual model that is used since it is strongly tied to the speaking context. Since we know that speaking time is a good indicator of dominance (Jayagopi et al.), it follows that the influence on the VFOA would lead to estimates which are closer to the context observed in the speech. We observe that the difference between the frame and event-based features for **RVA** is much bigger here compared to using the manual VFOA estimates, indicating the sensitivity of the features to automated estimates of the VFOA. We know that for the automated estimates of the VFOA, short glances tend to be missed and so this indicates that they are quite important to this feature. We also see that the features which are calculated based on the interpersonal influence have quite high performance for the I^{out} and ΔI . The interpersonal influence incoming (I^{in}) performed much worse. Also, as seen previously, the group-based VDR performs slightly better in general than the dyadic VDR measures.

If we compare the performance of the **RVA** and **GVDR** measures for the manual and automatically estimated VFOA, we see that while both frame-based measures perform better when using the automatic VFOA features, the performance decreases for the event-based measures. We can understand this better by observing the distribution of the amount of time that is used to calculate each measure. These are shown in Figure 11 and 12. We can observe that using the estimates of VFOA, the number of events that support the

measures is much less than for their corresponding measures using the manually annotated VFOA. In particular, the **RVA** feature uses even less events, which may also explain the larger decrease in performance when comparing this with the manual case.

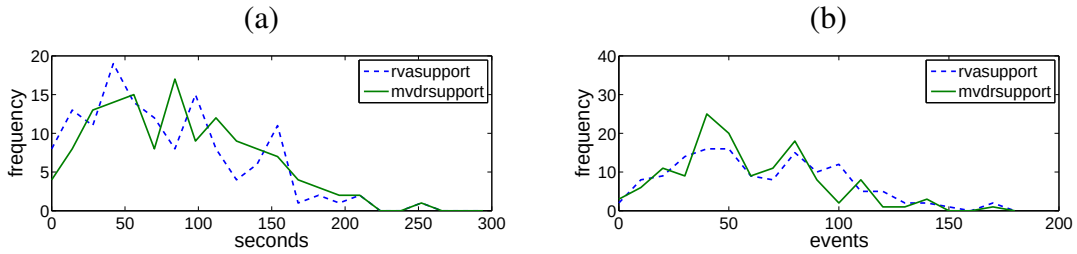

Figure 11: (a) Figure showing the distribution of the average amount of time per person, which is used to support the **RVA** and **GVDR** measures. Manual annotations of VFOA were used here. (b) This uses the events rather than frames to calculate the interval of support.

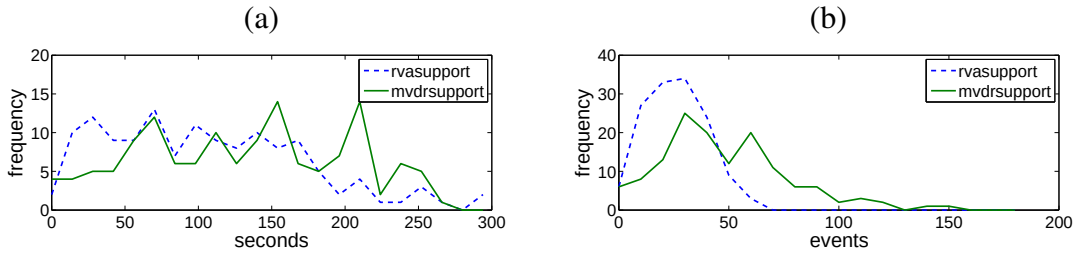

Figure 12: (a) Figure showing the distribution of the average amount of time per person, which is used to support the **RVA** and **GVDR** measures. Automatic estimates of VFOA were used here. (b) This uses the events rather than frames to calculate the interval of support.

5.2.7 Results: Estimating the Least Dominant Person

We also conducted experiments to estimate the least dominant person using the same visual dominance features. In this case, we evaluated the measures by taking the person with the smallest value to be the least dominant. For the I^{in} feature, the person with the highest value was taken to be the least dominant.

Again, we first observed the differences across all the visual dominance measurements when using the manual annotations of VFOA. These are summarised in the 'M' columns of Table 18. From these results we can see that the best performing feature was **RVA** and **GVDR**$^{\mathbf{N}}$ using the frame-based method with a classification accuracy of 80.7%. Using the event based method **RVA** and I^{out} performed well with a classification accuracy of 77.4%. **GVDR** was also among the better performing features with a classification accuracy of 67.7% and 64.5% for the frame-based method and event-based method respectively.

Estimating the Least Dominant from estimates of VFOA using head-pose information only

Least Dominant	Frame			Event		
VFOA Method	M	F	H	M	F	H
MostVotesPairVDR	45.2	56.5	**77.4**	48.4	59.7	67.7
MeanVDR	64.5	67.7	**77.4**	67.7	71.0	61.3
ProductVDR	64.5	61.3	61.3	61.3	61.3	54.8
GVDR	67.7	71.0	74.2	64.5	**74.2**	64.5
ΔGVD	48.4	71.0	58.1	51.6	71.0	61.3
GVDR^{N}	**80.6**	67.7	61.3	71.0	58.1	51.6
GVDR^{D}	25.8	51.6	38.7	22.6	48.4	29.0
RVA	**80.6**	**77.4**	45.2	**77.4**	67.7	29.0
SuccessfulFloorYield	n/a	n/a	n/a	34.9	36.3	33.1
MeanAudienceMonitor	n/a	n/a	n/a	32.3	17.2	20.4
I^{in}	67.7	61.3	48.4	64.5	69.4	58.1
I^{out}	77.4	**77.4**	71.0	**77.4**	**74.2**	**69.4**
ΔI	19.4	9.7	29.0	19.4	12.9	33.9

Table 18: Summary of results for the estimating the Least Dominant person when the method of estimating the VFOA is varied. M:Manual, F:Full context model, H:Head pose only. The best performance for each VFOA method is shown in bold.

Next, automatically estimated VFOA using head-pose information only was used to estimate the least dominant person. The results are summarised in the 'H' columns of Table 18. We can see here that the best performing method were **MostVotesPairVDR** ,**MeanVDR** ,**GVDR** and I^{out} .

Estimating the Least Dominant from context dependent Estimates of VFOA
Next, automatically estimated VFOA using context was used to estimate the least dominant person. The results are summarised in the 'F' columns of Table 18. From these results we can see that the best performing feature was **RVA** and I^{out} . **GVDR** and **ΔGVD** were some of the other best performing methods.

5.2.8 Comparing the Results Across Both Dominance Tasks

Event vs. Frame-based Features
In terms of events and frame-based features, we expected the event-based features to quantify better the ability for an individual to change or steer the conversation. This was certainly observed when the VFOA was annotated manually but not when automated VFOA estimates were used. This can be explained by the VFOA estimates being prone to errors when trying to estimate short glances.

Manual vs. Automatically Annotated VFOA
It was surprising to see how well the dominance estimation performed when automatically estimated VFOA features with contextual cues were used compared to the manual annotation case. As discussed previously, the increase in performance in some cases could be due to using the speaking context, which we know already to be a good estimator of dominance, to estimate the VFOA. However, for the automated VFOA estimated with no context, the performance tended to be worse.

Full context vs. Non-context-based estimates of VFOA
From the results from estimating the VFOA, we see that using no context to estimate the VFOA significantly affects the VFOA performance, with a 15% absolute decrease. In general, the full context method tended to give better dominance estimation performance than the no context VFOA estimates. The **RVA** feature showed the greatest sensitivity when using the two automated techniques.

5.2.9 Summary and Conclusion

In summary, we have found that using VFOA, it has been possible to estimate both dominant and non-dominant behaviour. Our best performing feature and estimation method for estimating the most dominant person used the group-based visual dominance ratio (**GVDR**), the full context model for estimating the VFOA and the frame-based features, given a performance of 82%. We found that this improved upon using the manual estimates of VFOA for estimating the most dominant person, probably because the full context model uses information from the speaking status to better estimate the VFOA. When estimating the least dominant person, we found that the best performing feature and estimation method could be found when using both the full context and head-pose only methods. For the head-pose only model, both the **MostVotesPairVDR** and **MeanVDR** methods worked well in the frame-based scenario, which could estimate the least dominant person correctly in 77% of the meetings. The I^{out} and **RVA** features performed similarly well for the full context model of VFOA estimation, attaining the same performance of 77% for estimating the least dominant person in the meetings. Our findings show that using the visual focus of attention to help estimate dominant behaviour did not surpass using speaking length as a single audio cue (85% for the most dominant and 84% for the least dominant task).

5.3 Analysing Cohesiveness

Cohesion in groups has been studied in social psychology from the perspective of military teams (Griffith [2007]), psychotherapy groups (Braaten [1991]), team sports (Carron et al. [1998]), task-based teams (Gammage et al. [2001]) and also social groups. For all but the last example, understanding cohesion in groups can help to improve the performance of the group but is not generally considered the main affect for good performance. However, it was found that good cohesion in military teams led to better performance under stressful situations. However, social and task-based cohesion are inextricably linked since being socially cohesive can help teams to be cohesive in task-oriented situations.

Many psychologists have tried to pinpoint what cohesion in groups means and research has shown that this can depend very much on the situation so that measures of cohesion among people in a group psychotherapy situation will be slightly different from a military or sport scenario. For analysing the AMI meeting data, we concentrate on group cohesion in the context of task-oriented scenarios. Griffith [2007] suggested that good cohesion leads to less concern about self welfare and more concern about attainment of group goals. Braaten [1991] suggested 5 factors that affect group cohesion in group psychotherapy: attraction and bonding, support and caring, listening and empathy, support and caring, self-disclosure and feedback, process performance and goal attainment.

Attraction and bonding, which is one of the areas related to task-oriented teams, encompasses factors related to a reasonable similarity of values, educational level, admiration, affiliation, belongingness, collaboration, community, compatibility, engagement, enthusiasm, motivation, sharing responsibility and solidarity. In addition, listening attentively and mutual stimulation through positive feedback were also listed.

Carron and Brawley [2000] rather defined group cohesion to be a group's resistance to disruption. They also suggested that groups could be socially very cohesive but lack task unity. They defined cohesion may be defined as "a dynamic process that is reflected in the tendency for a group to stick together and remain united in the pursuit of its instrumental objectives and /or for the satisfaction of member affective needs" (Carron et al. [1998] (p213)).

In terms of leadership and group cohesion, Siebold [1999] suggested that group leaders hold a group together and encourage a sense of pride in the group. He defined two axes to team cohesion, namely horizontal cohesion (related to peer bonding) and vertical cohesion (related to having a caring leader and also pride and also shared values, needs and goals within the group).

5.3.1 Annotation Procedure

A total of 21 annotators were used to annotate 120 2-minute non-overlapping meetings from the Idiap AMI meetings. 100 of the meeting slices were taken equally from each of the 10 groups who were asked to design remote controls. 20 meetings slices were taken from two other groups who were involved in real rather than scenario-based meetings. One involved discussing movies that could be shown at a film club and the second was a meeting to discuss the new allocation of offices to members of staff. Meetings were purposefully chosen so that the participants remained seated through the duration of the slice. The annotators were divided into groups of 3 and the meetings were organised such that each annotation group annotated 1 meeting from each of the teams in the data set, so 12 meeting slices per group. Some annotators belonged to more than one annotation group but no groups were identical.

To annotate the meetings for cohesion, terms used in the psychology literature were pooled together to create a questionnaire containing 27 questions in total. These included scoring the meetings based on how comfortable participants were, how integrated the team appeared, how well they knew each other, how engaged or involved they were, whether they shared the same goal, etc. For each question, annotators were asked to score their response on a 7-point scale. To ensure that annotators thought carefully about each of the questions, the valences of each answer were randomly flipped. If participants were unsure of the answer to any of the questions, they could leave the answer blank.

The annotators viewed their corresponding meeting slices through a web interface so that all the groups and meeting times were annonymised. Annotators who volunteered to annotate more than one set of 12 given one set at a time to minimise the bias of prior knowledge on teams that they had already seen before.

5.3.2 Analysing the Annotations

To analyse the agreement amongst annotators and across meetings and questions, we used the kappa agreement measure. Since the annotations scores were on a scale, we used the weighted kappa measure with a linear decay from the confusion matrix diagonal. From these kappa scores, we were able to observe the variation in kappa across different meetings or questions. Figure 13 shows the variation in kappa for all the questions. Overall, we see that the kappa agreement tends to be quite low. However, some of the questions appear to have higher agreement (questions 4-12 and 14, 15, 22, 24 and 25).

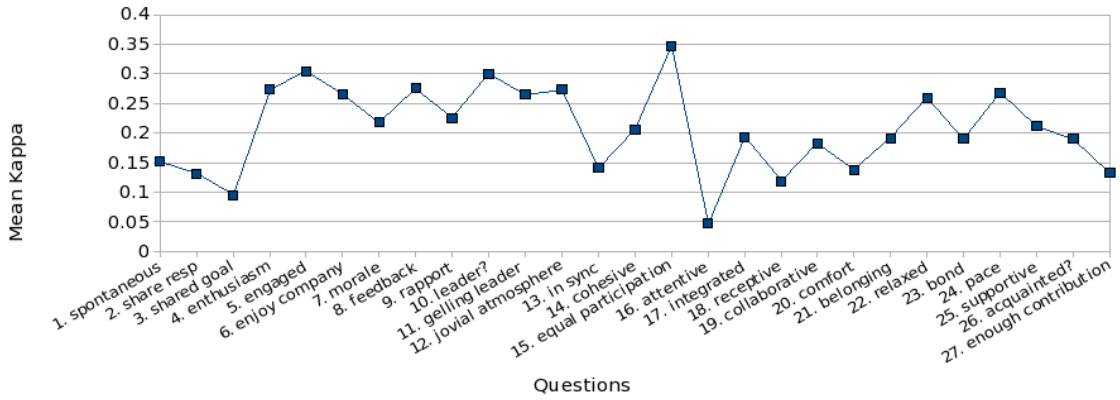

Figure 13: Mean kappa for each of the questions for all the annotators.

Next, we analysed the mean kappa values across annotation groups, as shown in Figure 14. Here, we see that the kappa values tended to be higher. This suggests that certain meetings have clearer cohesion characteristics than others and that when this was true, the annotators tended to have higher agreement. To confirm this, we analysed the effect of

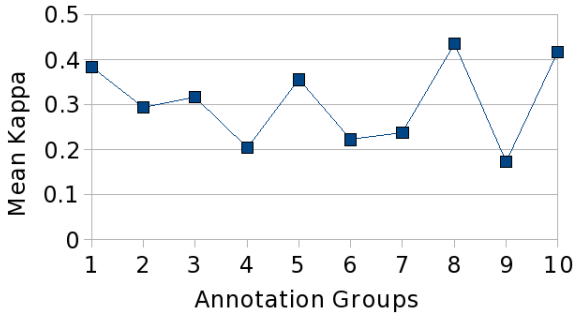

Figure 14: Mean kappa for each of the groups of annotators for all 12 meetings that they annotated.

the kappa agreement across all the annotated meetings to see if there was a relationship between the annotator scores for each question and the kappa agreement amongst annotators for the same meeting. Figure 15 shows how the scores varied as the kappa agreement increases from 0.4 upwards. The annotations from 42 meetings are shown which accounted for 5 and 4 meetings in kappa bands '>0.7' and '0.6-0.7' respectively and 17 and 16 meetings for the bottom two bands. The valences of the scores were flipped for

the cases in the questionnaire where the scores were high when the attribute was low. As we can see from Figure 15, the scores go closer to the central score of 4, as the kappa agreement decreases. This is probably due to the nature of the AMI data, which tends to be more collaborative and are carried out by volunteers and so it is unlikely that anyone would be purposefully uncollaborative. Observing the scores more closely, we see that in the top two kappa bands, there are no meetings with low cohesion while in the lower two bands, a few meetings exhibit low cohesion patterns. However, these account for only 7 meetings out of 33 so do not affect the average scores significantly. Figure 16(a) and (b) shows colour visualisations of the scores for the top kappa band ('>0.7') and also '0.4-0.5', which clearly show the decrease in agreement and also the low cohesion meetings. The visualisations are plotted with each meeting represented as a column of 3 (one for each annotator) and the questions are ordered by row.

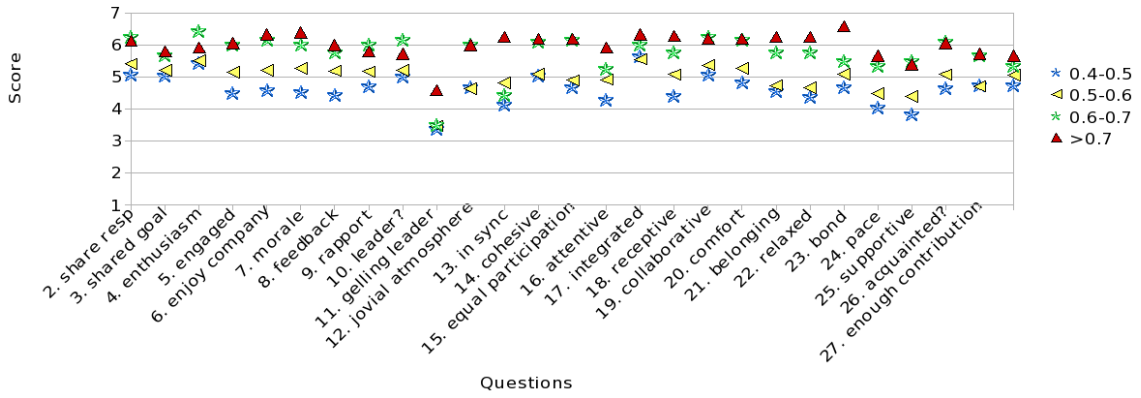

Figure 15: Graph showing the valence of the scores depending on the kappa agreement of the meetings.

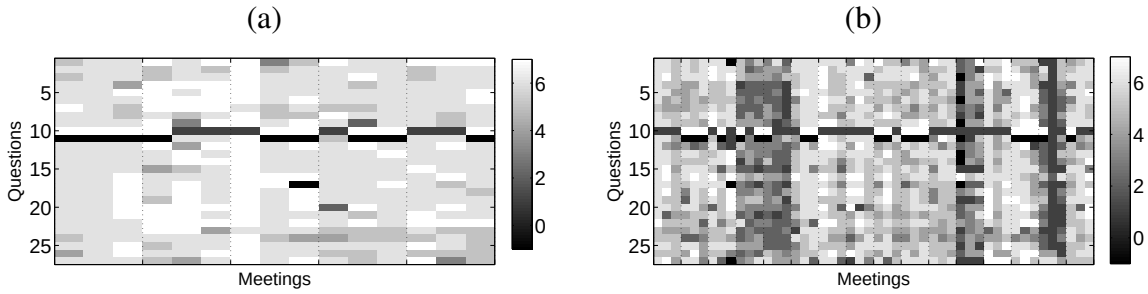

Figure 16: Colour visualisations of the annotator scorings for (a) meetings with a kappa agreement >0.7 (b) meetings with a kappa agreement between 0.4-0.5. All questions which the annotators felt they could not answer were marked with -1, otherwise the scores ranged from 1 to 7. Question 10 and 11 were highly correlated since 11 was answered depending on if the annotators thought there was a strong leader (question 10).

It was also interesting to observe from Figure 15 that when annotators were in high agreement, it tended to be for meetings where high scores could be given for all questions. Very

few meetings in the set where the kappa agreement was over 0.4 showed low cohesion characteristics. For question 10, since we only asked whether there was a leader or not, we labeled 'yes' to be 1 and or 7 to be 'no'. We see that for the meetings where the kappa agreement was greater than 0.7, there was less likely to be a leader in the group. However, for the cases where there was a leader the answers for question 11 showed that the leader tended to bring the group together. For the meetings where the kappa agreement was lower (less than 0.6), the average scores were one point lower.

Performing factor analysis on the annotations confirmed our findings that the meetings tended to show high rather than low cohesion. This was demonstrated by the strong separation of the variables or questions, depending on the orientation of their valences in the questionnaire. As would be expected, questions 10 and 11, which were about leadership showed strong correlation, having high factor loadings on the 3rd component. The factor loadings each question are shown in Figure 17. Since the factor loadings tend to separate each variable based on the orientation of the valence, this suggests that the data is biased towards meetings where there tends to be good cohesion.

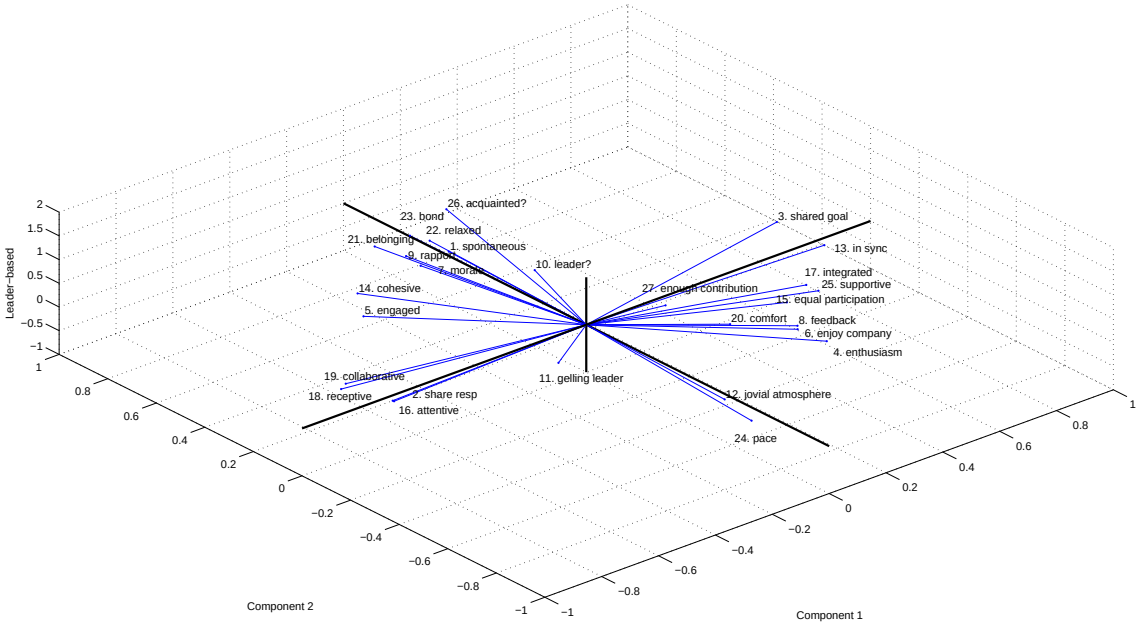

Figure 17: Factor loadings for each question. Questions 10 and 11 are clearly correlated. The rest of the questions are separated by the orientation of the valence for each of the questions.

Overall, our annotation analysis shows that our data is biased towards high cohesion meetings. We selected the meetings with the higher kappa agreement (above 0.4) for our experiments. From these 42 meetings, 35 could be categorised into the high cohesion category and the rest were labeled in the low cohesion group.

5.3.3 Cue Extraction

Audio cues were extracted by firstly automatically segmenting the audio signal from each headset microphone using the voice activity detection method of Dines et al. [2006]. From

this, various cues related to the speaking activity in the group was generated.

We designed the cues based roughly on the questions that we used in the questionnaire. They can be roughly summarised into 4 categories: periods between each individual's turns, times between floor exchanges, turn durations, and overlapping speech.

Pauses between individual turns
To quantify the degree of participation of each individual in the group, we use features related to the pause time between each person's turns. That is, the time between one person's turns.

- TPT: The total pause time between a person's turn and that same person's next turn. We would expect that during high cohesion meetings, there tends to be more equal participation among the participants so everyone will take a lot of turns but will also need to allow time for their fellow team-mates to talk. This feature encodes the amount of time spent actively engaged in a conversation without speaking.
- minTPT: Rather than taking the total time between a person's turns, this feature takes the minimum across all the team members. We would expect that this feature is more discriminative since every team-mate must be actively involved when not speaking, for this feature to have a high value.

Pauses between floor exchanges

- TS: The total silence time represents the pace of the conversation. If participants are distracted, uncomfortable with each other, or are not well acquainted, it is likely that there will be more periods of silence.
- MeanFE, MedianFE: The mean or median time between all floor exchanges. This encodes whether the pace of the conversation is fast or slow. Conversations at a fast pace will tend to have less time between floor exchanges.

Turn lengths

- MinTL: This is the minimum of each individual's average turn length. For highly involved conversations, we could expect that everyone will have relatively short equal average turn lengths. Taking the minimum ensures that all team members must be participating for the meeting to be considered at a high cohesion level.
- TT: The total number of turns made in the meeting. A turn is defined as a continuous period of time for which an individual has a speech activity values of 1.
- TSL: Total speaking length. We would expect the total speaking length to be higher for highly cohesive groups since there would be more activity in the meeting.

Overlapping speech

- TBC: The total number of short turns, or what what could be considered back-channels. This encodes all short turns (<4s) to be short turns. We would expect that highly cohesive would involve meetings where team members give each other more feedback. Therefore, there should be a higher number of back-channels.
- TOBC: This takes only the short turns which occur during the longer turn of someone else. We expect these to be more directly related to active feedback when someone is talking.

- TOT: The total overlap time encodes the amount of time that at least two people are speaking at the same time. We expect that more overlapping speech is due to conflict so the cohesion will tend to be lower.

5.3.4 Estimating High and Low Cohesion Meetings

Since little data was available, we used a simple algorithm to estimate whether a meeting was high or low cohesion. For each class, the average value for each feature was used and then a threshold was selected by taking the mean of the values in the high and low cohesion classes. To minimise problems with over-fitting the data, the high cohesion data was randomly sampled so that there was an equal number of data points in each class. Also, the experiments were carried out using a leave-one-out approach so that the test and training data were separate. Finally, for each feature and each test data point, the experiments were carried out 100 times to account for variations in the sampling process. The final performance is given as an average of these trials.

5.3.5 Experiment

The experiments were carried out as described in Section 5.3.4. Table 19 summarises the results. The second column indicates how the features were used; '+' means the feature is positively correlated with high cohesion. The '+' label is only shown when the same orientation was used consistently for all 100 trials and 42 test data points. The next two columns shows the average number of true positives. The bracketed number at the top of each column shows the number of data points in each class. The final column shows the average classification accuracy for each feature. The best performing feature was TPT

Features	Corr.	low cohesion (7)	high cohesion (35)	classification accuracy (%)
TPT	+	5.98	33	**92.8**
MinTPT	+	6	29.65	84.9
TT		3.98	27.19	74.2
MeanFE		2.77	20.36	55.1
MedianFE		3.82	28.61	77.2
MinTL	+	6	31.63	89.6
TSL		2.89	19.16	52.5
TBC		3.97	24.47	67.7
TOBC	+	6.16	24.13	72.1
TOT	+	5.56	27.67	79.1

Table 19: Summary of results run on 100 trials. Random performance would be 50% classification accuracy.

(with a classification accuracy of 93%) which encodes the amount of time each person spends not talking in between their turns. This feature is particularly interesting because it represents how attentive each team member is to the others in the group. The attentiveness can be shown through taking and discussing further a team member's ideas or providing many back-channels. This feature will have low values if one person tends to talk a lot while the others don't say anything. The second best performing feature was the minTL

feature which encodes the minimum average turn length. We would expect that in high cohesion meetings, everyone is participating a lot so the minimum average turn length will tend to be higher. For very uncohesive meetings, it could be that some people do not speak at all. The third best performing feature was the minTPT features, where we would again expect that in high cohesion meetings the minTPT tends to be higher. For all three of these features, the estimation of whether the meeting was high or low cohesion was consistently based on the feature value being above the threshold.

One of the worst performing features was the total speaking length, which also did not have a consistent orientation for the selected threshold. One would have expected the total speaking time to be quite discriminative. However, it seems to be that the TSL is not able to encode the level of interaction among the participants.

An interesting result was obtained with the TOT feature, which we expected would be negatively correlated with cohesion. However, it appears that more overlap is a reliable sign of high cohesion. When comparing the TOT performance with TOBC, we see that using just the number of overlapping back channels led to slightly worse performance. Therefore, the duration of overlap caused by successful interruptions appear to be correlated with high cohesion for our data. This is in keeping with some findings in social psychology that interruptions are indicative of good rapport when people are able to finish each other's sentences Tannen [1993].

5.3.6 Conclusion

We have shown some preliminary results on estimating cohesion from the AMI meetings using non-verbal cues. Unfortunately, due to the nature of the data, the meetings tended to be more in the high cohesion category when annotators were in agreement about the level of cohesion in the meeting. Our results show that it is possible to estimate low and high cohesion from our meeting data and that measuring the degree of attentiveness is as important as the amount of vocal participation in the meetings.

5.4 Predicting Remote vs. Collocated Group Interactions

Increasingly teams are expected to collaborate across different physical locations. The challenges involved in designing such remote collaboration systems are many- the communication infrastructure, the human-computer interfaces, improving the awareness of the participants, etc. The goal of such designs is to make remote meetings to be as close as possible to the face-to-face interactions.

The difference in the dynamics between collocated and remote meetings is significant and it can lead to poorer performance in distributed teams (Hinds and Bailey [2003]). Some of the difficulties of the remote participants and the group as a whole in remote meetings include the inability to conduct side conversations, the challenge of occupying the floor because of the lack of eye contact or the inability to utilize posture shifts and the phenomenon of in-room attendees forgetting about the remote people (Poel et al. [2008], op den Akker et al. [2009]).

Various technological approaches have been proposed to provide feedback to mitigate these differences (for example the meeting mediator (Kim et al. [2008])). Some approaches

allow real-time multimodal visualization of conversation analysis to improve the interactivity of group meetings (Otsuka et al. [2008], DiMicco et al. [2004], Poel et al. [2008]). Quantifying and measuring the difference in the group dynamics of collocated and remote meetings, using behavioral cues and more specifically nonverbal cues has been done in different ways in the literature although not extensively. In O'Conaill et al. [1993], video conferencing using ISDN system (with transmission lags, poor quality video), LIVE-NET (with less transmission lags, high quality video) and face-to-face interactions was compared. The differences were compared by studying back-channels, interruptions, turns etc (obtained using manual annotation). In Kim and Pentland [2009], the difference was captured by observing the groups with dominant people. When the group had one or more dominant people, it also had more speech overlap in remote meetings without any feedback. In contrast, much more work on characterizing the groups meeting face-to-face has been done (Gatica-Perez, Jayagopi et al. [2009]).

In this work we study two novel research questions, in the context of characterizing group dynamics in collocated and remote meetings. Firstly, can we distinguish between remote and collocated meetings? and secondly, can we predict the remote participant in the remote meetings? using only nonverbal cues. The nonverbal cues we consider in this work are based on acoustic information (speech activity based).

subsections 2 discusses our feature extraction process. subsection 3 introduces our experimental setup. subsection 4 documents the results obtained and subsection 5 gives the conclusions of our analysis.

5.4.1 Cue extraction

Nonverbal cues particularly audio based ones, are known to contain useful information for social inference - both vertical and horizontal aspects et al. [2005]. We extract the following nonverbal cues to characterize individual participants and the group as a whole.

Figure 18 shows extraction process and the associated tasks which are explained below.

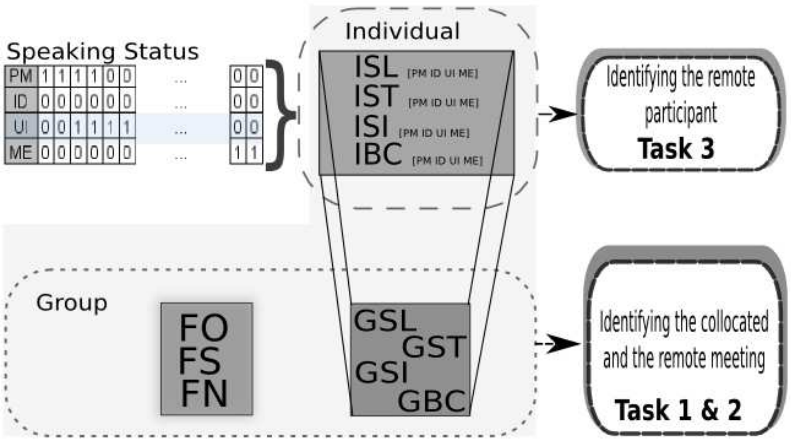

Figure 18: Our feature extraction process

We used the binary segmentation available with the data corpus (described in subsection Experimental Setup) - speech and non-speech for each of the participant. This is usually

computed by thresholding speaking energy values or using more sophisticated algorithms to combat cross-talks. A turn is a continuous period of time for which the person's speaking status is 1. Then we compute the features to characterize an individual and the group as a whole as follows (similar to the work Jayagopi et al. [2009]).

Individual features From the speech segmentation, we compute the following features.

- Speaking Length (ISL[i]): Considers the total time that a participant i speaks according to his speaking status.
- Speaking Turns (IST[i]): IST is the turns accumulated over the entire meeting for every participant i.
- Successful Interruptions (ISI[i]): The cumulative number of times that participant i starts talking while another participant j speaks, and i finishes his turn before j does, i.e. only interruptions that are successful are counted.
- Back-channels (IBC(i)): The cumulative number of times that participant i starts talking while another participant j speaks, and i finishes his turn before j does, i.e. only unsuccessful interruptions that are successful are counted.

Group features From the speaking status of all the participants, the following features to capture the overlap, silence patterns of a group as whole were computed. Let T be the total number of frames in a meeting, S be the number of frames when no participant speaks, M be the number of frames when there is a monologue and O be the number of frames when more than one participant talks.

- **Fraction of Overlapped Speech :** $FO = \frac{O}{T}$.
- **Fraction of Silence :** $FS = \frac{S}{T}$.
- **Fraction of Non-overlapped Speech :** $FN = \frac{M}{T}$.

Additionally we compute Group Speaking Length (GSL), Group Speaking Turns (GST) and Group Successful Interruptions (GSI) which are accumulated over all the participants.

5.4.2 Experimental Setup

Dataset: The Augmented Multi-Party Interaction with Distance Access (AMIDA) corpus et al. [2007] consists of 10 hours of recorded, transcribed and annotated meetings recorded at the University of Edinburgh. The meeting data has a similar character to the scenario data in the Augmented Multi-Party Interaction (AMI) Meeting Corpus, but the AMIDA corpus contains meetings with one remote participant. Recordings were gathered using 24 microphones (two-circular arrays of eight, four headset and four lapel mic), six cameras (four close-up, two view of the room-center of the table and corner) and output from a slide projector. There are three four-person meetings (for a total of 27-meetings) of which two have a remote participant (18-meetings). Figure 20 shows the scenarios of the AMIDA corpus.

Meetings: The 9 sets of participants in the AMIDA meetings are involved in the design of a new remote control and meet at least three times.

Meeting A - New Project Start: In this meeting participants decide collectively on role allocation (who should do what), and discuss the aim of the project.

Meeting B - Conceptual Design: This meeting consists of individuals presenting their work and the group coming up with a conceptual design via videoconferencing.

Meeting C - Detailed Design: During this meeting participants present their individual work and, present and evaluate the clay prototype again via videoconferencing.

Every meeting has a participant with one of the following roles- project manager (PM), industrial designer(ID), marketing expert (ME) or a user interface designer (UD). In all the remote meetings the user interface designer is the remote participant (as shown in figure 20). Figure 19 shows the meeting rooms and a sample remote meeting.

Figure 19: Top: The meeting room of collocated participants (left) and the meeting room of the remote participant (right). The two monitors in each of the room show the rest of the group members. Bottom: The meeting view that the remote participant (left) and the collocated participants look at during the meetings (right)

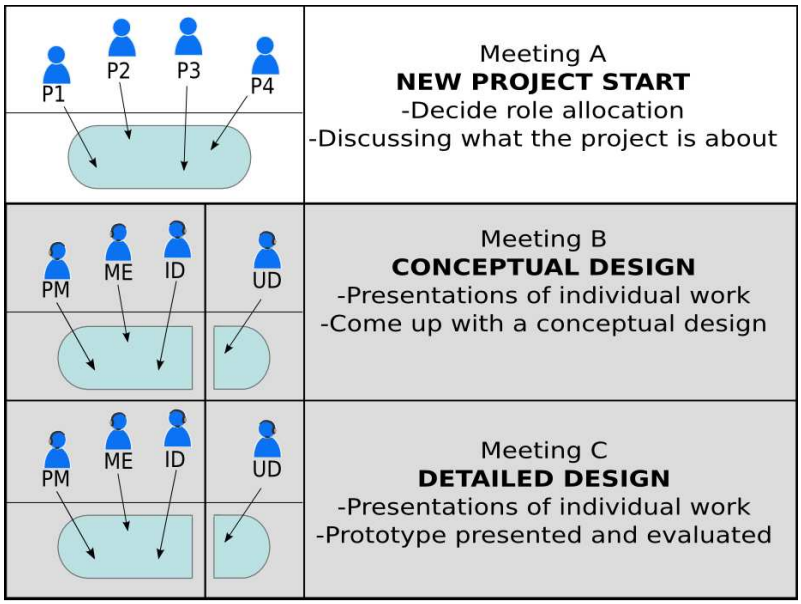

Figure 20: The scenarios of the AMIDA corpus. UD is the remote participant in B and C meetings.

We use the meetings from the AMIDA corpus for studying the group interactions in remote versus collocated settings. Average duration for collocated, also called A-meetings is 18.57 minutes, for remote meetings (called B and C meetings, see figure 20) average

duration is 37.7 minutes. While the collocated meeting was a pure discussion type meeting, the remote meeting had presentations followed by discussions. In order to have fair comparison, we consider only the last five minutes of B and C meetings (which mostly were discussions) for our subsequent analysis .

Tasks: In order to model the difference between collocated and remote meetings, we define three tasks.

Task 1: The first task is to distinguish between collocated and remote meetings. For this classification task, A meetings belong to first class and, B and C meetings belong to the second class.
Task 2: Goal of the second task is infer the collocated meeting given three meetings (1 collocated and 2 remote), where the participants are the same. This task is simpler when compared to the first task, as it assumes that it is only possible to identify a collocated meeting versus a remote meeting composed of the same participants.
Task 3: The third task is to predict the remote participant in the remote meetings.

5.4.3 Results

Task 1: For the first task, we learnt a Gaussian mixture model using Expectation Maximization (EM) algorithm Bishop [2006] for each of the group features. Table 20 shows cross-validation performance of this task with one and two Gaussians. For this task, the Group Speaking Turns (GST) with 2 gaussians had the best performance (70%) which is slightly above random performance (66%), but not statistically significant (given the fact that the size of the dataset is small). Figure 21 suggests that while learning a global threshold to classify collocated and remote meetings might be difficult (due to inter-group variations), making a local decision (by looking at only the meetings of same participants) would still be possible. The results of task 2 verify that this indeed is true.

Gaussian - 1		**Gaussian - 2**	
Features	**Accuracy**	**Features**	**Accuracy**
GSL	62%	GSL	52%
GST	**66%**	GST	**70%**
GSI	62%	GSI	57%
FN	59%	FN	53%
FS	62%	FS	50%
FO	62%	FO	39%
Random	66%	Random	66%

Table 20: Performance of group features on predicting the collocated and remote meeting (Task 1).

Task 2: For the second task, we have only 3 meetings with the same participants (9 such sets). Therefore, we use a simple unsupervised approach to predict the collocated meetings - hypothesizing that collocated meetings have either the minimum or the minimum value of the group feature. Table 21 shows average performance given group features. For this task the Group Speaking Turns (GST) again performed the best (81% accuracy),

showing that the collocated meeting have lesser number of turns as compared to the remote meetings. One possible reason to explain this is that because of the presence of a remote participant the group need more turns per unit time to achieve their desired objective. Also, this result is statistically significant (p = 0.01) compared with random performance (33%). From Figure 21 we can observe that collocated meetings are the ones with less speaking turns.

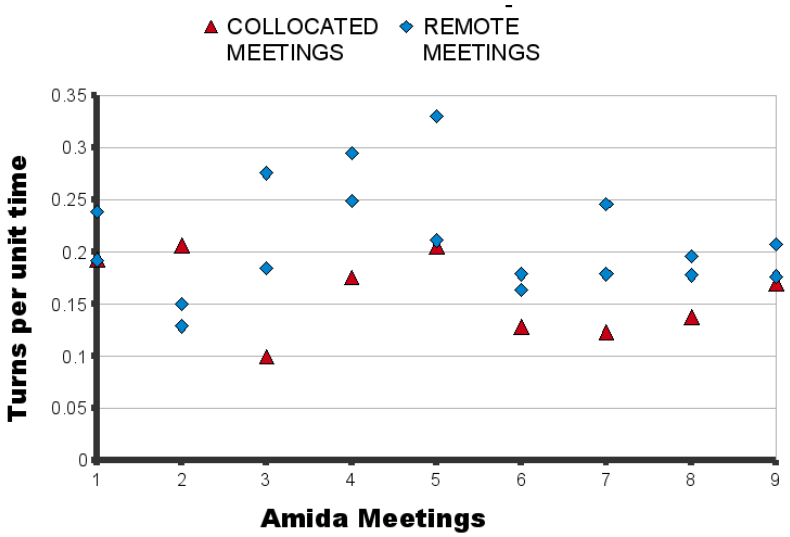

Figure 21: Group Speaking Turns for each of the 9 sets of AMIDA meetings.

Minimum		Maximum	
Features	**Accuracy**	**Features**	**Accuracy**
GSL	44%	GSL	15%
GST	**81%**	GST	11%
GSI	33%	GSI	7%
GBC	22%	GBC	44%
FN	22%	FN	67%
FS	28%	FS	33%
FO	22%	FO	22%
Random	33%	Random	33%

Table 21: Performance of group features on predicting the collocated meeting (Task 2).

Task 3: For the task of predicting the remote participant in a meeting, we hypothesized that the remote participant has either the minimum or the maximum value of individual features. Table 22 shows the results. Minimum value of speaking length (ISL) is the individual feature that better predicts who is the remote participant (50% accuracy, not statistically significant), given that random performance corresponds to 25% accuracy. Figure 22 shows average speaking length per role on remote meetings, as we can see the role with less speaking length is the user interface designer(UD) followed closely by the industrial designer(ID). The scenarios of the AMIDA corpus make this task even more challenging.

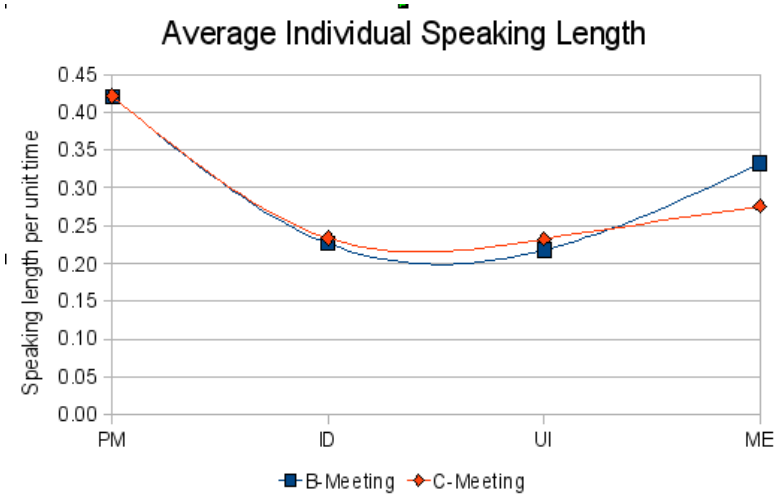

Figure 22: Average of Individual Speaking Length for each of the roles in remote meetings

Minimum		**Maximum**	
Features	**Accuracy**	**Features**	**Accuracy**
ISL	**50%**	ISL	17%
IST	28%	IST	33%
ISI	46%	ISI	11%
IBC	28%	IBC	33%
Random	25%	Random	25%

Table 22: Performance of individual features on predicting the remote participant (Task 3).

5.4.4 Conclusions

In this work, we attempted to characterize the differences in group dynamics between collocated and remote meetings on a very recent data that is publicly available (the AMIDA corpus), by computing speech activity based nonverbal cues. We evaluated the effectiveness of these cues in predicting the remote versus collocated meetings and the remote participant. Based on the results we noticed that collocated meetings have less turns and the remote participant talks less. We understand that such conclusions are limited given the size of our dataset. Further, it is noteworthy that the AMIDA corpus was not collected with the task of classifying remote and collocated meetings in mind. Therefore the scenarios in these meetings are different, making our task a challenging one. In the future, we would like to expand our dataset to study the generalisability of our findings and extract more relational and visual features to improve our performance on the three tasks. Such 'learned' classifiers could be used to evaluate the technologies that aim at mitigating the differences between face-to-face and distributed meetings.

6 Summarization and Paraphrasing

As in previous years, we distinguish between *abstractive* and *extractive* meeting summarization where abstractive summaries use a symbolic representation of meeting and summary contents to generate novel sentences, while extractive methods summarize a meeting by detecting the most salient segments of the transcript and quoting those parts verbatim.

In addition to these principal summarization approaches, we report on two new lines of research, *argument diagramming* and *participant profiling*. The last section of this chapter is devoted to our ongoing research in the field of automatic *paraphrasing*.

6.1 Abstractive Summarization

During the last year, we worked on three different parts of our abstractive summarization approach in parallel—representation, interpretation, and generation.

6.1.1 Representation

We continued our efforts from the second year of the AMIDA project to drive the development of an ontology-based representation of the remote control design scenario. Using the COnAn annotation tool [cf. AMIDA Deliverable 5.4], we progressed to identify and integrate the central concepts and relations to represent AMI and AMIDA meetings.

We argue that the domain dependence inherent in knowledge-based symbolic approaches can be met by separating our core summarization engine from the knowledge bases it operates on. To assert that our ontological commitments allow to exchange the remote control design domain against other scenarios, we have integrated our ontology with the *DOLCE Lite+* upper model [Gangemi et al., 2002] as we have proposed before.

The *Description and Situation* (DnS) [Gangemi and Mika, 2003] part of *Dolce Lite+* introduces a model for embedding ontological entities into specific situations. Situations are represented as ontological entities themselves. By introducing such a special entity we have the means to technically refer to complex contexts through a first order symbol ("reification"). We make use of the DnS reification framework to address a number of linguistic issues that would otherwise be difficult to realize ontologically. For instance, our new model has correspondences to *tense*, *modality*, and *pluralities*.

On a technical level, our ontology is realized in the semantic web language OWL[10]. In our processing pipeline, it can be seen as the backbone in terms of data structures which are passed between the interpretation, transformation, and generation modules.

6.1.2 Interpretation

One of the most challenging aspects in the processing pipeline for abstractive summarization is the interpretation of meeting contents. We concentrate on the richest source of information, the meeting transcript.

10. `http://www.w3.org/TR/owl-ref/`

With an extension of our previous approach using the SPIN language interpretation system [Engel, 2006], we can define simple patterns of lexical constructs without the need to use a fully specified derivation tree or a grammar. For a transcript of spontaneously spoken language, even when discarding ASR errors and speech disfluencies, this greatly simplifies the interpretation process. Subsequent application of multiple rules allows the recognition of complex language patterns, it also reduces the combinatorial complexity of the matching process by factoring out common sub-matches. For example, each of the following three rules is used to identify when a speaker references a remote control in which case an ontology instance is created to represent this remote control:

```
remote -> RemoteControl();
remote control -> RemoteControl();
controller -> RemoteControl();
```

Now, future rules can refer to this object instead of the original surface forms.

```
a yellow $R=RemoteControl() -> $R(color:yellow)
```

This particular rule refines the ontological representation by adding a `color` role with a value `yellow`. The overall interpretation process successively applies such matching rules to the dialog act segments of the transcript.

Besides refining the SPIN rule base and adapting it to the changes brought about by the new ontology, another novel feature we added during the last year was the possibility to refer to certain context information within the matching rules. By way of example, consider the following discourse snippet, adapted from AMI meeting ES2003a.

```
A: Uh, my name is Dave Cochrane.
B: And you're going to be the the user interface designer.
```

This part of the meeting is summarized in the gold-standard manual summary of ES2003a as follows: *The team members introduced themselves to each other by name and by their roles in the project.* An ontological representation must reflect the occurrence of *introducing* perdurants in the above dialog, but we note that there is a difference between the two dialog acts. In the first dialog act, speaker `A` refers to himself while in the second dialog act, `B` introduces the role of `A`. To account for such difference, we require special context-dependent construct in our matching rules.

```
my name is $N=Name()
-> Introducing(agent:@getContext("speaker"),
               about:$N(name-of:@getContext("speaker")));

you're [going to be] the $R:Role()
-> Introducing(agent:@getContext("speaker"),
               about:$R(played-by:@getContext("addressee")));
```

In the second rule, we assign the matched role not to the current speaker, but to the person addressed by the current speaker [cf. e.g. Jovanović, 2007].

Our method to deriving the necessary rules still requires manual labor. Although the different statements of a gold-standard summary in the corpus are linked to those parts of the transcript that contain relevant information to support the summary statement, this link annotation is too imprecise to be exploited in automatic ways through, e.g., a supervised learning approach. It is, for example, not obvious which specific part of the summary

System	Progr. Lang.	Documentation	Ease of use / Complexity	Purpose	Limitations
ASTROGEN http://people.dsv.su.se/~hercules/ASTROGEN/ASTROGEN.html	Prolog	Examples	Easy/ Medium	NLG for beginners	Cases of objects
FUF/SURGE http://www.cs.bgu.ac.il/surge/index.html	Lisp	Examples + Scientific papers	Medium/ High	General Purpose	n/a
MUG http://www.david-reitter.com/compling/mug/	Prolog	Tutorial, Paper	Hard/ Medium	GUI prompts	n/a
NipsGen Available per DFKI	Java	Manual, example	Medium/ High	Dialog systems	n/a
SimpleNLG http://www.csd.abdn.ac.uk/~ereiter/simplenlg	Java	API, manual, tutorial	Very easy/ Low	Small auxiliary program	Complex grammars

Table 23: An overview of the evaluated NLG systems.

statement is supported in the linked transcript segment. On the other end of the scale, fine-grained manual annotation of full transcripts with symbolic representations proved tedious if not infeasible.

The employed method balances between the above possibilities. We asked annotators to read the full transcript of a meeting and insert comments summarizing the ongoing events at a granularity of about 5-25 dialog acts. As a result, the transcripts are split into smaller paragraphs each of which is described with a short summary sentence. In a second step, a domain expert transferred the purely textual representation into ontological structures. The different parts of the structures are then mapped to the utterances in the transcript paragraph which evoke them–this mapping is lastly expressed in the form of multiple SPIN rules as described above.

Despite the obvious amenity of this method, it will be a focus of future research how to minimize or even eliminate the required manual work through automatic means.

6.1.3 Generation

For the textual generation of summaries, we re-evaluated existing state-of-the-art natural language generation (NLG) systems to find the best suited candidate. Table 23 lists the evaluated systems. In addition, the following systems were initially considered but eventually dismissed for the reasons provided. Note that one of our requirements is that the system has to be platform-independent because of our heterogeneous development environment.

System	Reason for Omission
CLINT http://www.cs.bgu.ac.il/~elhadad/clint.html	Microsoft Windows only
Concordance http://www.concordancesoftware.co.uk	Commercial, MS Windows only
GenI http://trac.loria.fr/~geni	Installation failed, documentation scarce
KPML http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/README.html	MS Windows only
SPUD http://www.cs.rutgers.edu/~mdstone/nlg.html	Installation failed, no documentation available
TG2/Themsis http://www.dfki.de/~busemann/more-tg2.html	No software available

The examined systems differ in the complexity as well as in the application spectrum they are intended for. Some of them are clearly tailored towards rather simple purposes (e.g. SimpleNLG), yet we found some of them disproportionally complicated to use (e.g. MUG).

Overall, FUF/SURGE and NipsGen proved to be the best choices. They both provided acceptable complexity and comprehensive capabilities. In the end we opted for NipsGen since we had prior experience with the system from the AMI project which facilitated the transition of the required resources (grammar files, production rules, knowledge base). Also, since it is written in the Java programming language—as opposed to Lisp in the case of FUF/SURGE—it is easier to integrate with the overall summarization system which is developed in Java as well.

A general description of NipsGen can be found in [Engel and Sonntag, 2007]. For summarization, we had to adapt the internal type system of NipsGen to be in concordance with our ontology. Dynamic mapping from ontological concepts to typed feature structures (TFS) allows to conceptualize domain entities and relations in input structures for the generator. Linguistic types are separated from types of the remote control design domain. To be able to generate sentences that are similar in style to manually written summary sentences, we drew inspiration from the gold-standard summaries available in the AMI corpus: by re-generating sentences as they appear in these summaries, we were able to identify which entities to include in our domain ontology and to create the necessary set of generation rules for NipsGen at the same time. As a result of this top-down approach, we now have at our disposal a generation system that can express complex syntactic constructions from a relatively simple input structure which is derived with no or only little modifications from our ontological representation of meeting contents.

6.2 Presentation of Decision-Based Summaries

We have also developed an integrated system to extract and present the decision-related parts of the AMIDA meetings. The system first recognizes proposals and the ensuing (dis-)agreements, determines acceptance or rejection decisions and represents them in a domain-specific knowledge model (ontology). A discourse memory collates the decisions and keeps track of the current state of discourse, resulting in a representation of all valid decisions at the end of the meeting, including links into the discourse, representing the decision process, see **?** but also AMIDA deliverable D5.4, **??** for related work. Finally, we have updated an existing presentation system (SuVi, see AMIDA deliverable D5.4) to summarize the relevant discourse and visualize it in a storyboard style. We are currently conducting an extrinsic evaluation that compares our results with previous work on summarization and decision detection.

Figure 23: Two decision segments.

Figure 24: Storyboard-based summary, also available through a web-based browser that includes links into the meeting transcript and the video stream.

6.3 Extractive Summarization

Our work on extractive meeting summarization in the last year focused on three areas: extension of the Integer Linear Programming based approach we proposed last year **?** with sentence in addition to concept level information modeling, investigation of graph based methods, and investigation of prosodic features in addition to lexical and structure related features and appropriate normalization of prosodic features.

The work on using Integer Linear Programming (ILP) for selecting an optimal set of sentences that maximize concept weights resulted in very good improvements in ROUGE-2 compared to the greedy MMR algorithm. We developed two models for ILP summarization, based on sentences and concept level information modeling. The first one corresponds to an optimal version of MMR and the second one can model redundancy in sets of multiple sentences instead of being limited to pairwise comparisons. We ran extensive experiments with different sentence similarity measures, different type of concepts (such as word n-grams, and key phrases we extract) and according to a wide range of length constraints. The main conclusions are that the concept-based framework (concept-ILP) is both more scalable and performs better than the sentence-based (MMR-ILP) framework. The sentence-level algorithms give lower rouge score because of their inability to properly model redundancy (by only considering pairs of sentences, not larger groups).

In our following work, we also studied methods for combining sentence and concept level scoring, and benefited from the combination **?**. Even though the concept-based summarization approach that we have developed yields very good performance, it is very sensitive to the quality of the input sentences and tends to select ill-formed short sentences to pad summaries with more concepts. We want to replace the current adhoc filtering of short sentences and sentences with less concepts with a more robust and more justified approach. For that, we have extended our Integer Linear Program (ILP) decoder to handle sentence-level scores in addition to concept-level scores. The new method uses a linear interpolation between concept and sentence scores, which can be unbalanced by difference in length between sentences. We have investigated multiple normalization methods for transforming sentence-level scores so that they are compatible with the concept-level score space. This new framework lead to significant improvements in term of ROUGE score over using concepts alone, especially in the case of automatic speech recognition output.

We also proposed a new algorithm, ClusterRank, that is based on graph-based methods (such as PageRank and TextRank), and uses a clustering step to group consecutive utterances that are about the same topic for meeting summarization **?**. Our experiments showed that clustering significantly improves the summarization performance in comparison with the PageRank and TextRank algorithms for both manual transcriptions and automatic speech recognition output.

Most of the previous work on meeting summarization focuses on lexical features, whereas previous studies on speech summarization benefited from the use of prosodic features. In the last year, we also worked on extracting prosodic features from meeting utterances, and using them in supervised methods, in addition to lexical and structure related features. We extracted several prosodic (such as pitch, energy and duration related) features, and studied normalization of prosodic features for speaker, topic and fixed window seg-

ments. The prosodic features resulted in higher performance than using lexical features only, especially on speech recognizer output, but their combination resulted in the best performance **?**

6.4 Argument Diagramming

Our work in the year also included utterance type classification for argument diagramming, for estimating the flow of reasoning in meeting discussions. For that purpose, we used the Twente Argumentation Schema (TAS) annotations on AMI meetings. We used multi-class classification for this task and have shown that both lexical and prosodic features are useful **?**.

6.5 Participant profiling

In the majority of meetings, individual participants perform one or more different roles (both at a meeting level as well as an organisational level). For example, in the AMI corpus scenario meetings, each participant's organisational roles are well defined: a project manager (PM), an industrial designer (ID), a marketing expert (ME) and user interface (UI) designer. Furthermore, the PM also performs the role of chair within the meeting.

To fulfill a particular role, a participant will exhibit a range of skills and be able to draw upon previous experience. Such skills and experience may be different from other roles within the meeting. Furthermore, the topics of interest within the meeting will vary according to the participant's role. Such role knowledge could allow meeting summaries to be tailored to a particular participant's role or allow participants with particular skill sets and interests to be indentified from a corpus of meetings.

A participant profile can be represented in a number of different formats and be generated in different ways. For example, the profile could consist of a small number of keywords or key whole phrases extracted from the meeting transcript. Such a profile does not necessarily have to be a generic summary; it could characterize the individual in terms of how they interact with other participants both on a short timescale (portions of a meeting) through to much longer timescales (many meetings). If extended further, the profile could encapsulate the participant's interests, motivations, opinions and group roles (eg, marketing) over these timescales.

This work concentrated on assessing the possibility of extracting group role information based upon techniques similar to TF*IDF. In order to investigate the feasibility of this, we attempted to find correlations between various metrics extracted from the transcripts and the role of the person associated with that portion of the transcript. Metrics have included named entities (various combinations of ARTEFACT, COLOUR, ENAMEX, MATERIALS, SHAPE), key phrases, TF*IDF, etc. This work looks at the corpus in terms of roles; in the case of AMIDA's scenario meetings, this splits the corpus into four (reflecting the four roles PM, ME, ID and UI). It was found that these standard metrics did not highlight words or phrases which were specific to a particular role. Indeed, given the low number of roles a TF*IDF score becomes unreliable / difficult to interpret.

Therefore, an approach which took the role directly into account was chosen: an analysis of these four role groups which could identify words which are commonly spoken

by a particular role member but infrequently by members of other roles. Our new metric (RTF*INRTF - role term frequency * inverse role term frequency) assigns each word in the corpus one score per role. Each score relates to that word's distinctiveness with respect to that role. Although inspired by the TF*IDF equation in the form used by Sanderson [1996], RTF*INRTF has to take account of the potentially low number of roles (c.f. usually high numbers of documents when using TF*IDF).

R is defined as the set of all roles. In this case $R = \{PM, ID, UI, ME\}$. RTF is the frequency of a term spoken by a participant in a specific role across the whole corpus:

$$RTF_{r,w} = \frac{ln(n_{rw} + 1)}{ln \sum_k n_{rk}} \tag{23}$$

where n_{rw} is the number of times term w is uttered by a participant in role r where $r \in R$.

INRTF is the inverse frequency of a term spoken by any participant in a role other than r:

$$INRTF_{r,w} = \frac{ln \sum_{p \in R, p \neq r} \sum_k n_{pk}}{ln \sum_{p \in R, p \neq r} (n_{pw} + 1)} \tag{24}$$

Therefore, words with a high RTF*INRTF will have a high importance to a specific role. Given these scores, it is possible to generate extractive summaries for each participant (who has a particular role) in each meeting. Current work is focussing on determining both the consistency within roles and the distinctiveness between roles. We are considering both criteria derived from automatic evaluation techniques (e.g., BLEU or ROUGE) as well as summary comparison techniques using human subjects as judges. When considering automatic evaluation criteria, the lack of a 'gold-standard' summary must be considered: we wish to compare a number of summaries, all of which have been generated automatically. Another aspect we are looking at concerns the data sparsity within the AMIDA corpus. For example, on average, over 1200 words are unique to each role. This clearly impacts upon any role-specific metric.

Currently, these profiling technologies use hand transcripts together with any relevant manual annotations, such as speaker ID and named entities. Future work will investigate the impact of replacing the manually created annotations with AMIDA ASR transcripts and automatically identified speakers and named entities. The focus in this later stage would be to measure the decline of the profile quality due to erroneous features and investigate approaches which will ameliorate such reductions in profile quality.

6.6 Paraphrasing

Much of the work on the Mutaphrase algorithm in the last year has been under the hood, in components that are not directly visible to the user. The major added component is a generalized unification routine that combines two input objects into a composite object which is consistent with both sources of information according to a flexible specification. This was necessary in a number of places in the algorithm, but most importantly this enables the algorithm to correctly deal with optional sentential elements, producing a much larger number of potential paraphrases and to more robustly produce sentences that follow

the rules of grammar (such as verbs having the right form to go with their subjects). Another added component, traversing the frame semantic hierarchy, allows the Mutaphraser to produce a broader set of paraphrases based on more distantly related semantic material. Although these improvements appear to reduce the number of illformed mutaphrases, the results on language model perplexity showed only insignificant improvement. We are in the process of analyzing these results.

We made major progress on integration of automatic frame parsers by using the 'lth' labeler from Lund University's Center for Applied Software Research. Although the lth parser does not output GFs (grammatical functions), the Mutaphrase algorithm is robust to their lack. The PTs (phrase types) as output by the lth parser are also somewhat problematic, though as reported below, the automatic parsing still produced good results.

We have implemented the mutaphrase algorithm to the point that it is capable of producing mutaphrases of simple input sentences. The system produces a large number of mutaphrases (up to 300,000 for some sentences) with a variety of phrase types and sentence orders. Some 28 % of the output are still 'bad' in terms of well-formedness (estimated by hand-counting 300 mutaphrases of real-world sentences). Even in the simplified input sentence 'I gave presents to friends', the system outputs 96 (syntactically filtered) mutaphrases, of which 9 are 'bad'. A random selection from the output includes 'I handed in presents', and 'I handed the present over to friends'.

We then performed several experiments to measure the efficacy of the mutaphrase algorithm. We use language model perplexity as a metric. First, the NTI corpus was divided into two halves by selecting sentences at random. The first half was designated as the training set. The second half was designated as the in-domain test set. A completely independent set of sentences chosen from the PropBank corpus was designated as the out-of-domain test set.

Next, a language model was generated from the training set (without mutaphrases). Then, mutaphrases were generated for each sentence in the training set using manually annotated semantic role labels and also automatically generated semantic role labels using the lth parser. For each sentence, 100 mutaphrases were sampled and an ngram language model was generated from the mutaphrases. Finally, the two language models were combined using a standard optimal linear interpolation method. In the end we showed that adding automatic mutaphrases to the NTI corpus vs adding manual mutaphrases only made a difference of less than 2 % in perplexity for the difficult out-of-domain case, despite the fact that many mutaphrases are 'bad'. With improvements to the mutaphrase algorithm, we should only expect even larger gains. Also notice that despite the fact that the lth parser does not output GFs, the PTs are somewhat problematic, and the errors inherent in automatic parsing, the final results of using the automatic parser on the out of domain test set are only slightly worse than using the manual role labels.

7 Meeting Profiler

The AMIDA project has developed a Meeting Profiler within WP5. This meeting profiler enables the user to quickly view what topics were addressed during the meeting and at what time, by displaying a set of evolving tag clouds of important terms on a timeline. These terms were extracted with term extraction algorithms from the transcribed meeting speech. As such, the browser constitutes a form of *extractive* summarization (Murray et al. [2005]), and is particularly relevant to on-line and remote settings, where users want a quick update of an ongoing or past (missed) meeting. In the next subsections, we describe the technology and functionality of this browser. In Section 8, we describe an extension that allows for *abstractive* summarization of meetings.

7.1 Functionality of the Meeting Profiler

The Meeting Profiler provides a windowed view on a meeting; its interface is shown in Figure 1. The tool is web-based; it runs in a browser. A drop-down button on the top left,

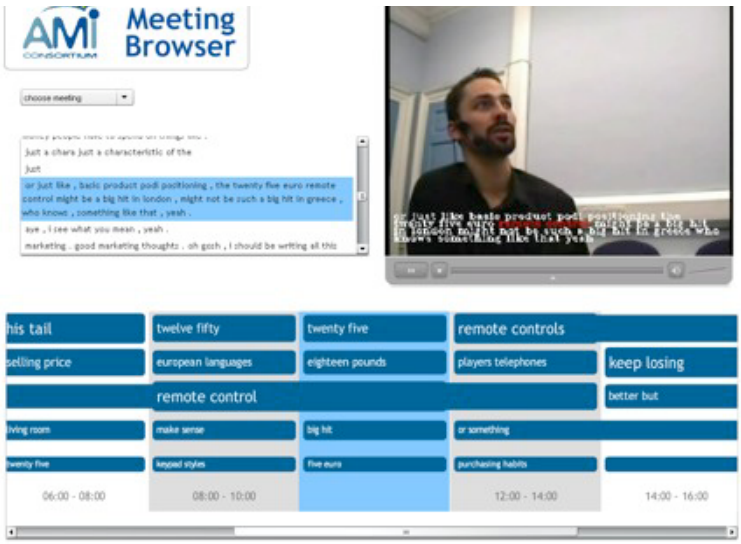

Figure 25: Interface of the Meeting Profiler.

allows for the selection of a meeting. The corresponding metadata of the meeting is read from an XML file and displayed in the user interface. On the right hand side is the video window, containing the footage of the selected meeting. The user can choose to play the video from the start, or browse through the video with the fast forward/backward button, or by clicking on the timeline. The transcript in the transcript window (left of the video) is aligned with the chosen point in time in the video. The Tagcloud window (below) always highlights the tag cloud corresponding with the point in time in the video. The tagcloud visualisation of the meeting allows for quick browsing through the meeting. It consists of a number of tag clouds, one for each 2-minute segment. Clicking on a cloud plays the video from that point in time, and aligns the transcript accordingly. For each segment, the five most descriptive terms (uni- or bigrams) are chosen. The size of the font indicates the relative importance of the term with respect to the other terms in that segment. The

tag cloud that corresponds to the video fragment currently playing is highlighted in blue. The user can also browse through the transcript. Clicking in the transcript starts the video from exactly that point.

Users can click on a term in a tag cloud to jump to the right moment in the video recording of the meeting to view what was said. The topics are displayed in tag clouds on a timeline. The user can view and interact with this timeline in order to get a quick impression of the topics discussed in the meeting.

7.2 Tag Cloud Generation

The following steps generate the tag cloud timeline:

1. Split the meeting transcript into equal pieces (windows) of a predefined length.
2. Extract the most salient terms from each window using language models
3. Select the top N terms to be placed on the timeline.

The first step is straightforward; split the transcript into windows using the timecodes in the transcript files. Window size of 2 to 3 minutes were found to be optimal, in the sense of having adequate resolution and length for proper term extraction. Term extraction was performed using a background unigram model reconstructed from a general spoken English trigram model. Unigram counts were estimated from this corpus using Heap's Law (van Leijenhorst and van der Weide [2005]). Experiments with a similar model estimated from general written English demonstrated an advantage of the spoken corpus. Unigram, bigram and trigram term weights were computed with an interpolated variant of TF.IDF. For instance, the term weight of a unigram t, $w(t)$, was computed as follows:

$$w(t) = tf_F(t) \cdot \log\left(\frac{(\lambda \cdot tf_F(t)) + (1-\lambda) \cdot tf_B(t)}{tf_B(t)}\right) \tag{25}$$

with $tf_F(t)$ the frequency of a term t in a foreground corpus (in our case consisting of 2-3 minutes of meeting speech), and $tf_B(t)$ the frequency of t in the background corpus. The interpolation parameter λ was varied, and finally set to 0.9, and we displayed only the 5 topmost salient items. Additional background terms were manually added as stopwords. We compared visualizations of only bigrams with visualizations of uni-, bi- and trigrams, and settled on the bigram version, being far more informative from a user perspective than unigrams, and less prone to inaccuracies than trigrams.

7.3 Video

The video that is produced for the visualization of a meeting takes as input all available footage, consisting of close-up videos of every participant, and an overall global view video of the meeting. A specially designed automatic *video producer* makes an intelligent decision as to which video stream to mix in the final mix, depending on who is speaking. The final video is synchronized with the text transcript of the meeting.

The text transcript of the meeting provides speaker identification combined with the time (or frame number) when somebody is talking. If somebody is speaking, the automatic video producer takes the close-up video of that person as the current output. When nobody is speaking the global overview video is taken. To circumvent continuous swapping

between different person close-ups, the view is only changed to a person when he or she is going to say a significant amount of words (in general: more than 'yeah', 'no' and 'hmm').

Besides swapping videos to close-up of persons speaking, the video producer can also subtitle the video with the text transcript. Additionally in the subtitles, salient terms from the tag cloud can be highlighted in a different color to emphasize them.

8 Cross-Lingual Abstractive Summarization

In this section (published as Raaijmakers et al. [2009]), we describe our submission to the VideoCLEF'09 Linking Task. This cross-lingual task consists of cross-linking speech-recognized Dutch TV speech to English Wikipedia pages. Approaches to this problem yield technology that is directly applicable to the Meeting Profiler: being able to link raw content (speech) to Wikipedia information yields a complementary facility of *abstractive* summarization, as opposed to the *extractive* summarization described above.

Our system consists of a weighted combination of off-the-shelf and proprietary modules, including the Wikipedia Miner toolkit of the University of Waikato. Using this cocktail of largely off-the-shelf technology allows for setting a baseline for future approaches to this task.

8.1 Introduction

The Finding Related Resources or linking task of VideoCLEF'09 consists of relating Dutch automatically transcribed TV speech to English Wikipedia content. For a total of 45 video episodes, a total of 165 anchors (speech transcripts) have to be linked to related Wikipedia articles. Technology emerging from this task will contribute to a better understanding of Dutch video for non-native speakers.

The AMIDA approach to this problem consists of a cocktail of off-the-shelf techniques. Central to our approach is the use of the Wikipedia Miner toolkit developed by researchers at the University of Waikato[11] (see Milne and Witten [2008]). The so-called *Wikifier* functionality of the toolkit detects Wikipedia topics from raw text, and generates cross-links from input text to a relevance-ranked list of Wikipedia pages.

We investigated two possible options for bridging the gap between Dutch input text and English Wikipedia pages: translating queries to English prior to the detection of Wikipedia topics, and translating Wikipedia topics detected in Dutch texts to English Wikipedia topics. In the latter case, the use of Wikipedia allows for an abstraction of raw queries to Wikipedia topics, for which the translation process in theory is less complicated and error prone. Specific to our approach is a weighted combination of various modules, and the use of a specially developed part-of-speech tagger for uncapitalized speech transcripts.

11. See `http://wikipedia-miner.sourceforge.net`

8.2 System setup

In this section, we describe the setup of our system. In subsections 8.3, 8.4 and 8.5, we describe the essential ingredients of our system. In subsection 8.6, we define a number of linking strategies based on these basic ingredients, which are combined into scenarios for our runs (section 8.7).

8.3 From Dutch to English

For the translation of Dutch text to English (and following Adafre and de Rijke [2006]), we used the Yahoo! BabelFish translation service[12]. An example of the output of this service is given in Figure 26.

Figure 26: The result of Babelfish for a sample query.

Since people, organizations and locations often have entries in Wikipedia, accurate proper name detection is important for this task. Erroneous translation to English of Dutch names (e.g. 'Frans Hals' becoming 'French Neck') should be avoided. Proper name detection prior to translation allows for exempting the detected names from translation. A complicating factor is formed by the fact that the transcribed speech in the various broadcastings is in lowercase, which makes the recognition of proper names challenging, since important capitalization features can no longer be used. In order to address this problem, we trained a maximum entropy part-of-speech tagger: an instance of the Stanford tagger[13] (see Toutanova and Manning [2000b]). The tagger was trained on a 700K part-of-speech tagged corpus of Dutch, after having decapitalized the training data. The feature space consists of a 5-cell bidirectional window addressing part-of-speech ambiguities and prefix and suffix features up to a size of 3.

12. http://babelfish.yahoo.com/
13. http://nlp.stanford.edu/software/tagger.shtml

8.4 A Dutch Wikifier

The imperfect English translation by Babel Fish was observed to be the main reason for erroneous Wikify results. In order to omit the translation step, we ported the English Wikifier of the Wikipedia Miner toolkit to Dutch, for which we used the Dutch Wikipedia dump and Perl scripts provided by developers of the Wikipedia Miner toolkit. The resulting Dutch Wikifier ('NL Wikifier' in Figure 28) has exactly the same functionality as the English version, but unfortunately contains a lot less pages than the English version (a factor 6 less). Even so, the translation process now is narrowed down to translating detected Wikipedia topics (the output of the Dutch Wikify step) to English Wikipedia topics. For the latter, we implemented a simple database facility (to which we shall refer with 'The English Topic Finder') that uses the cross-lingual links between topics in the Wikipedia database for carrying out the translation of Dutch topics to English topics.

An example of the output of the English and Dutch Wikifiers for the query in Figure 26 is given in Figure 27. The different rankings of the various detected topics are represented

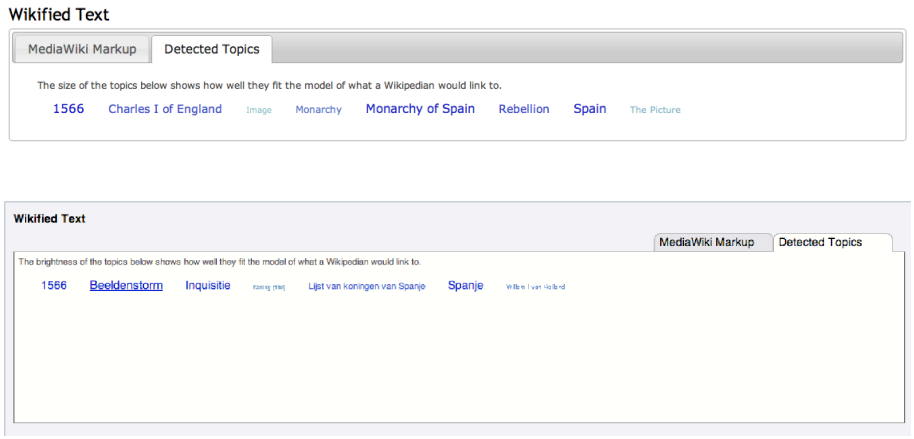

Figure 27: The result of the English and Dutch Wikifiers for a sample query.

as a tag cloud with different font sizes, and can be extracted as numerical scores from the output.

8.5 Text retrieval

In order to be able to entirely by-pass the Wikipedia Miner toolkit, we deployed the Lucene search engine (Hatcher and Gospodnetic [2004]) for performing the matching of raw, translated text with Wikipedia pages. Lucene was used to index the Dutch Wikipedia with the standard Lucene indexing options. Dutch speech transcripts were simply provided to Lucene as a a disjunctive (OR) query, with Lucene returning the best matching Dutch Wikipedia pages for the query. The HTML of these pages was subsequently parsed in order to extract the English Wikipedia page references (which are indicated in Wikipedia, whenever present).

8.6 Linking strategies

The set of techniques just described leads to a total of four basic linking strategies.

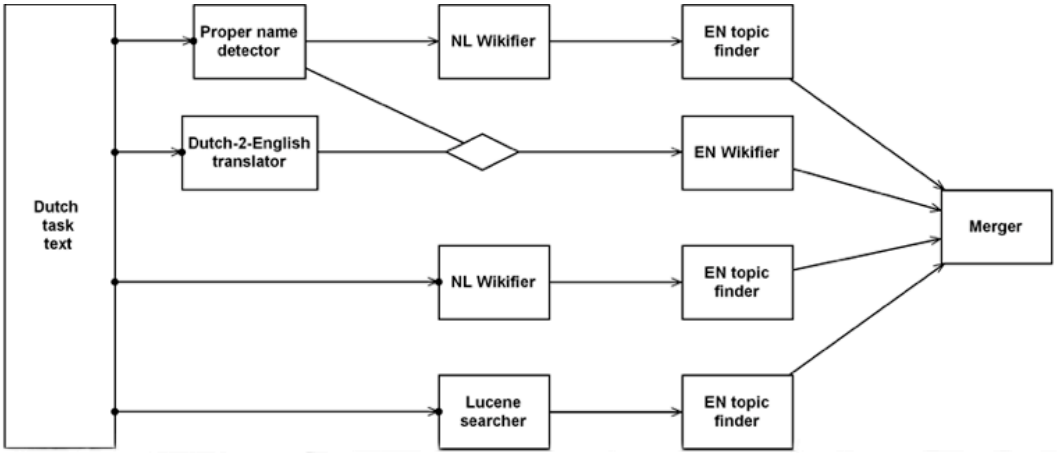

Figure 28: Cross-lingual linking setup.

Of the various combinatorial possibilities of these strategies, we selected 5 combinations for our submitted runs. The basic linking strategies are:

Strategy 1: proper names only (the top row in Figure 28) Following proper name recognition, a quasi-document is created that only consists of all recognized proper names. The Dutch Wikify tool is used to produce a ranked list of Dutch Wikipedia pages for this quasi-document. Subsequently, the topics of these pages are linked to English Wikipedia pages with the English Topic Finder.

Strategy 2: proper names preservation (second row in Figure 28) Dutch text is translated to English with Babelfish. Any proper names found in the part-of-speech tagged Dutch text are added to the translated text as untranslated text, after which the English Wikifier is applied, producing a ranked list of matching Wikipedia pages.

Strategy 3: topic to topic linking (3rd row from the top in Figure 28) The original Dutch text is wikified using the Dutch Wikify tool, producing a ranked list of Wikipedia pages. The topics of these pages are subsequently linked to English Wikipedia pages with the English Topic finder.

Strategy 4: text to page linking (bottom row in Figure 28) Lucene matches queries with Dutch Wikipedia pages. The English topic finder tries to find the corresponding English Wikipedia pages for the Dutch topics in the pages returned by Lucene. This strategy omits the use of the Wikifier and was used as a fall-back option, if none of the other modules delivered a result.

A thresholded merging algorithm removes any results below an estimated threshold and blends the remaining results into a single ordered list of Wikipedia topics, using estimated weights for the various sources of these results. Several different merging techniques were used for different runs; these will be discussed in subsection 8.7.

8.7 Run scenarios

In this section, we describe the configurations of the 5 runs we submitted. We were specifically interested in the effect of proper name recognition, the relative contributions of the Dutch and English Wikifiers, and the effect of full-text Babelfish translation as compared to a topic-to-topic translation approach.

Run 1

All four linking strategies were used to produce the first run. A weighted merger ('Merger' in Figure 28) was used to merge the results from the different strategies. The merger works as follows:

1. English Wikipedia pages referring to proper names are uniformly ranked before all other results.
2. The rankings produced by the second linking strategy ($rank_{EN}$) and third linking strategy ($rank_{DU}$) for any returned Wikipedia page p are combined according to the following scheme:

$$rank(p) = ((rank_{EN}(p) * 0.2) + (rank_{DU}(p) * 0.8)) * 1.4 \quad (26)$$

   The Dutch score was found to be more relevant than the English one (hence the 0.8 vs. 0.2 weights). The sum of the Dutch and English score is boosted with an additional factor of 1.4, awarding the fact that both linking strategies come up with the same result.
3. Pages found by linking strategy 2 but not by linking strategy 3 are added to the result and their ranking score is boosted with a factor of 1.1.
4. Pages found by linking strategy 3 but not by linking strategy 2 are added to the result (but their ranking score is not boosted).
5. If linking strategies 1 to 3 did not produce results, the results of linking strategy 4 are added to the result.

Run 2

Run 2 is the same as run 1 with the exception that linking strategy 1 is left out (no proper name recognition).

Run 3

Run 3 is similar to run 1, but does not boost results at the merging stage, and averages the rankings of the second and third linking strategy. This means that the weights used by the merger in run 1 (0.8, 0.2 and 1.4) are resp. 0.5, 0.5 and 1.0 for this run.

Run 4

Run 4 only uses linking strategy 1 and 3. This means that no translation from Dutch to English is performed. In the result set, the Wikipedia pages returned by linking strategy 1 are ordered before the results from linking strategy 2.

Run 5

Run 5 uses all linking strategies except linking strategy 1 (it omits proper name detection). In this run a different merging strategy is used:

1. If linking strategy 2 produces any results, add those to the final result set and then stop.
2. If linking strategy 2 produces no results, but linking strategy 3 does, add those to the final result and stop.
3. If none of the preceding linking strategies produces any results, add the results from linking strategy 3 to the final result set.

8.8 Results and discussion

For VideoCLEF'09, two groups submitted runs for the linking task: Dublin City University and the AMIDA concortium (TNO). Two evaluation methods were applied by the task organizers to the submitted results. A team of assessors first achieved consensus on a primary link (the most important or descriptive Wikipedia article), with a minimum consensus among 3 people. All queries in each submitted run were scored for Mean Reciprocal Rank[14] for this primary link, as well as for recall. Subsequently, the annotators agreed on a set or related resources that necessarily included the primary link, in addition to secondary relevant links (minimum consensus of one person). Since this list of secondary links is non-exhaustive, for this measure only MRR is reported, and not recall.

Run	Recall	MRR
1	0.345	0.23
2	0.333	0.215
3	0.352	0.251
4	0.267	0.182
5	0.285	0.197
Average TNO	0.32	0.215

Table 24: Recall and MRR for the primary link evaluation by TNO. (Average DCU scores were 0.21 and 0.14, resp.)

As it turns out, the unweighted combination of results (run 3) outperforms all other runs, followed by the thresholded combination (run 1). This indicates that the weights in the merging step are suboptimal. Omitting proper name recognition results in a noticeable drop of performance under both evaluation methods, underlining the importance of proper names for this task.

In addition to the recall and MRR scores, the assessment team distributed the graded relevance scores (see Kekäläinen and Järvelin [2002]) assigned to all queries. In Figure 29, we plotted the difference per query of the obtained averaged relevance score with

14. For a response $r = r_1, \ldots, r_Q$ to a ranking task, the MRR would be $MRR = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{rank_i}$, with $rank_i$ the rank of answer r_i with respect to the correct answer.

Run	MRR
1	0.46
2	0.428
3	0.484
4	0.392
5	0.368
Average TNO	0.43

Table 25: MRR for the secondary link evaluation by TNO. (Average DCU score was 0.21.)

the total average obtained relevance scores for both the Dublin City University (DCU) and TNO runs. For every video, we averaged the relevance scores of the hits reported by DCU and TNO. Subsequently, for every TNO run, we averaged relevance scores for every query, and measured the difference with the averaged DCU and TNO runs. It can be clearly seen that Run 1 and 3 obtain the best results, producing only a small amount of queries below the mean. Most of the relevance results obtained from these runs are around the mean, showing that from the perspective of relevance quality, our best runs produce average results.

8.9 Conclusions

In this contribution, we have taken a technological and off-the-shelf-oriented approach to the problem of linking Dutch transcripts to English Wikipedia pages. Using a blend of commonly available software resources (Babelfish, the Waikato Wikipedia Miner Toolkit, Lucene, and the Stanford maximum entropy part-of-speech tagger), we demonstrated that an unweighted combination produces competitive results. We hope to have demonstrated that this low-entry approach can be used as a baseline level that can inspire future approaches to this problem. A more accurate estimation of weights for the contribution of several sources of information can be carried out in future benchmarks, now that the VideoClef annotators have produced ground truth ranking data.

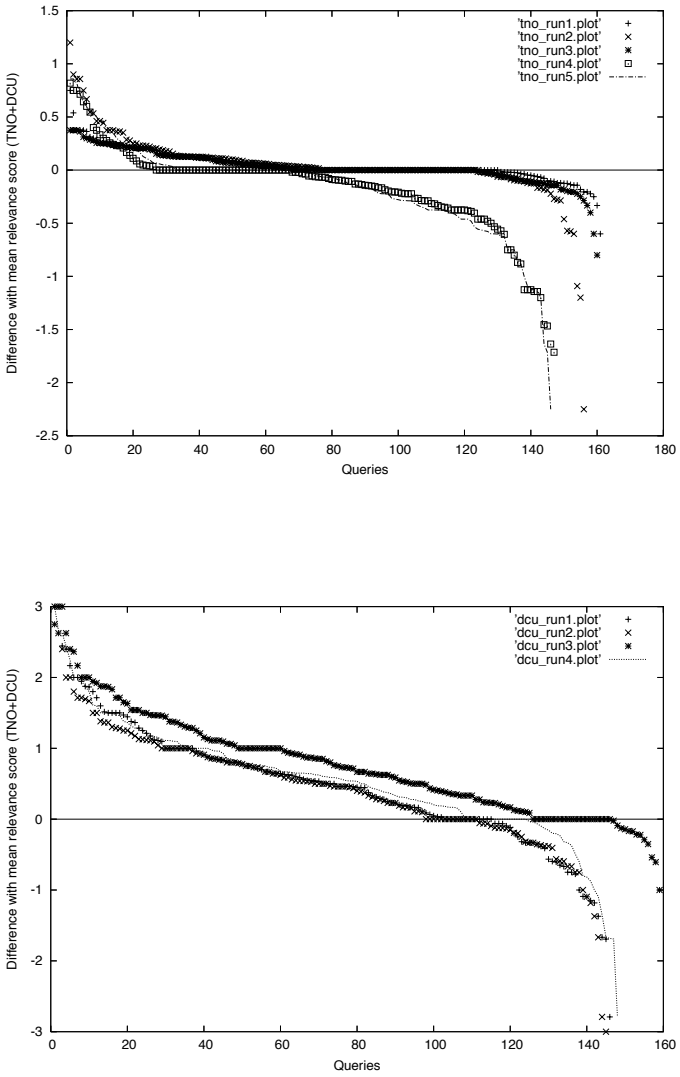

Figure 29: Difference plots of the various TNO (top figure) and DCU (bottom figure) runs compared to the averaged relevance scores of DCU and TNO.

9 Automatic Video Editing

Today's business world is full of meetings and of travel to meetings around the world. Video conferencing, as shown in Sabri and Prasada [1985], is a successful approach to reduce costs for companies. At the beginning of online conferencing only one video stream was exchanged between two locations. Nowadays different locations with multiple cameras are connected and a new problem arises: Which camera should be shown? Which cameras could be ignored?

Not only for online video conferences, but also for previously recorded meetings, it is an interesting topic to show a selected camera, which contains the most relevant pieces of information from the meeting. For the playback of past meetings a meeting browser, for example Wellner et al. [2004], can be used. These are the main usage scenarios of the systems which are described in upcoming sections.

9.1 Online Video Editing

In recent years the system for automatic and realtime video editing has been developed and improved. Its description together with description of the involved HUB interface can be found in foregone AMIDA deliverable documents (D5.2 and D5.4) and also in Herout et al. [2008]. In the last year we have focused on the realtime and online capabilities of mentioned system, that means other subsystems had to be added - one for handling input from several cameras and the other for the feature extraction.

Having several input streams, some robust and speed-efficient processing needs to be done to identify each stream importance, so that the video editing could be done. We have involved two own modules - one for the human head pose detection (based on Adaboost which details can be found in current AMIDA deliverable D4.5) and the other for motion-velocity detection (to discover quick changes in view - possibly interesting for the observer). These modules provides events, that are next used for the proper output stream selection. This selection is done in rule based manner, so new video editing rule configuration was added to the system database to reflect new types of events.

The input streams for the system comes from several firewire cameras. If the system runs on QuadCore CPU with at least 1G memory, up to six streams can be processed in resolution 640x480 with a frame rate above 16 frames per second.

The final system is capable of real online video editing, that is still open for other improvements or add-ons - new feature extraction modules. It can be used in any meeting room without specific camera setup. A demonstration video presenting the system and showing the video editing on real data can be seen on Youtubechannel [2009].

9.2 Offline Video Editing

Previous work concentrates on two different approaches for the offline scenarios. The first approach done by Sumec [2004] uses high level features, such as speech transcripts and person movements, and the camera selection process is based on rules. For the evaluation, people have been watching the created video and judged the quality. Therefore, it is impossible to compare it with others. The second approach uses low level features and two

Figure 30: Online video editing system setup with firewire cameras.

different models for the camera selection. The first model is based on thresholds, which can be seen in Al-Hames et al. [2006] and the second one uses Hidden Markov Models presented by Al-Hames et al. [2007].

During the last year of AMIDA we combined the best of both approaches to achieve better results. In the first step low level features are extracted from the audio and video sources. Additionally to these features, high level ones, such as group action, person action and person speaking, are used for the camera selection task. The second step is concatenating the features on feature level. After that, a segmentation and classification is done by Hidden Markov Models (HMM), which are described in Rabiner [1989]. Different combinations of features and settings for the models have been evaluated.

9.2.1 Features

Three different modalities of features are used: acoustic, visual and semantic. The first two modalities are low level features and are derived directly from the audio- and video streams. The well known MFCCs and Global Motions have been used for these two modaluties (For more details see Fang et al. [2001] and Wallhoff et al. [2004]). The semantic features contain more related information of the occurrences in the ongoing meeting. These features are interesting because of the close relation between what a person or the group is doing and which camera is important. The features, which have been applied are group action, person action and person speaking. In the following paragraphs the semantic features are described in more details.

The group action has been deeply investigated in the research community over the last couple of years, for example by et al. [2006] or Reiter et al. [2007]. The systems are working directly on audio and video streams and achieve reliable results, but they are currently not real time capable. The meeting is segmented into a sequence of labels like monologue participant one to four, discussion, presentation, whiteboard and note taking.

Moreover, a person action detection system has been developed by Zobl et al. [2003]. These systems create a sequence of actions for each of the participants, thus four features for each time frame are available. The labels used, are similar to the group actions but

contain some more classes: sitting down, standing up, nodding, shaking the head, writing, pointing, using a computer, giving a presentation, writing on the white-board, manipulation of an important item and idle. Idle for example is used if the person is speaking or listening to the meeting. The classes nodding or shaking should help to find points in the meeting where a person should be shown even though he is not speaking.

The last semantic feature which is currently used is the person speaking. It is a four dimensional vector which contains binary information for each participant and each time frame. The bits are set to one if a person is speaking.

9.2.2 Experiments

For all the experiments, a six-fold cross validation with person disjoint test and training sets were performed. Three different measurements are used for the evaluation: recognition rate (RR), action error rate (AER) and frame error rate (FER). High rates of RR are good and in the case of AER and FER lower values are better.

The experiments consist of two different tasks: classification and combined segmentation and classification. For the first one, the class boundaries are given and only a classification of these segments is performed. The results of this task are measured as RR and it is equal to the number of correct classified segments divided by the total number of segments. The second experiment is the combined process of finding the right class boundaries and classify these segments correctly. This is the real task of the system and the measurements for that are the FER and AER. The FER counts all the correct detected frames and divides them by the total number. Thus, the FER takes into account the correct position of the boundaries. The AER consideres only the correct sequence of segments.

In table 26, first the results of all single modalities are presented. The low level audio features achieve a FER of 50.1%, as the best single modality. Only the person speaking features performs nearly comparable. The visual features alone are not enough for the camera selection task, because most of the time the person who is speaking is important. The high AER of the low level features means that too many shot changes have been added to the video.

The first idea was combining acoustic features, such as audio or person speaking, with visual hints, such as global motion or person actions. The fusion of audio and group actions improves the results slightly. The use of person actions reduces the FER about 7.6% to a rate of 42.5%. This is already better than the best low level feature result achiedved during year three of AMIDA (44.6%) and all evaluated fusions in Al-Hames et al. [2007] (47.9%). The FER can be further reduced by combining all semantic features to a rate of 39.6%. The best results achieves a multi-stream HMM by using audio features and all high level features with a FER of 38.1%. For the RR and AER the picture is very similar, only for the RR the best model uses all semantic features only.

9.2.3 Conclusion

In this work we presented the combination of low level and semantic features for camera selection. The system performs a feature fusion using single and multi-stream HHMs. There is an reduction of 6.5% for the FER from the best low level feature model to the best

Table 26: Evaluation of different modality combinations. The number of states per model varies over the different combinations of modalities. MS indicates that a multi-stream model achieves this result. AER means action error rate, FER is the frame error rate and RR stands for recognition rate.

Model	AER	FER	RR
Audio (A)	158.7	50.1	47.6
Global Motion (GM)	177.5	64.3	34.8
Skinblob (SK)	600.3	78.6	16.8
Group Action (GA)	84.8	61.0	26.2
Person Action (PA)	72.2	62.8	28.2
Person Speaking (PS)	62.2	51.5	48.3
Audio & GA	63.1	49.2	48.3
Audio & PA	60.2	42.5	51.5
GA & PA & PS	58.3	39.6	54.8
A & GA & PA & PS (MS)	56.2	38.1	53.9
Audio & GM (MS)	60.8	44.6	52.9

combination. The integration of semantic features, such as group action, person action and person speaking, into the system is successful.

10 Conclusions

While the final year was mostly concerned with improving and finalizing on-going work, a number of new research topics have been explored. Many of the algorithms developed over the last year and the whole period of the AMI and AMIDA projects are now available and used in various settings, including two demonstration systems, the "content linking demo" and the "user engagement and floor control demo." Research is being applied more and more to other corpora and executed in other frameworks, e.g., the work in VideoCLEF'09.

As the underlying data are publicly available through the AMI-corpus, we expect to see more international research teams to pick up on the data, the tasks and the algorithms developed in AMIDA–a process that has already started in previous years. In these cases, AMIDA results are setting the standard and will certainly be improved by future research in the corresponding communities, including the AMIDA partners.

References

Rieks op den Akker. Meeting IS1003d - with annotator agreement analysis of AMI dialogue act and addressing annotation. *Internal Report - Department of Human Media Interaction, University of Twente.*

Elizabeth Shriberg. Preliminaries to a theory of speech disfluencies. *PhD thesis, University of California at Berkeley, 1994.*

Sebastian Germesin, Tilman Becker, and Peter Poller. Hybrid multi-step disfluency detection. In Proceedings of MLMI*, 2008.*

F. Jorgensen. The effects of disfluency detection in parsing spoken language. In NODALIDA*, pages 240–244, 2007.*

Jana Besser. A Corpus-Based Approach to the Classification and Correction of Disfluencies in Spontaneous Speech, January 2006.

X. Phan. Crftagger: Crf english pos tagger. 2006.

Y. Liu, E. Shriberg, A. Stolcke, and M. Harper. Comparing hmm, maximum entropy and conditional random fields for disfluency detection. In Eurospeech*, pages 3313–3316, 2005.*

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

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

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

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

S. Germesin and T. Wilson. Agreement detection in multiparty conversation. In Proceedings of ICMI-MLMI*, 2009.*

K. Toutanova and C. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In Proceedings of EMNLP/VLC-2000*, 2000a.*

D. Hillard, M. Ostendorf, and E. Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In Proceedings of HLT/NAACL*, 2003.*

T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of HLT-EMNLP-2005*, 2005.*

Natascha Korolija. Episodes in talk: Constructing coherence in multiparty conversation. *PhD thesis, Linköping University, The Tema Institute, Department of Communications Studies, 1998.*

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

J. Niekrasz and J. Moore. Participant subjectivity and involvement as a basis for discourse segmentation. In Proceedings of SIGDIAL*, 2009.*

Wallace L. Chafe, editor. The Pear Stories: Cognitive, Cultural, and Linguistic Aspects of Narrative Production*, volume 3 of* Advances in Discourse Processes*. Ablex, Norwood, NJ, 1980.*

Rebecca J. Passonneau and Diane J. Litman. Discourse segmentation by human and automated means. Computational Linguistics*, 23(1):103–139, 1997.*

Massimo Poesio and Mijail A. Kabadjov. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In Proceedings of LREC*, 2004.*

Doug Beeferman, Adam Berger, and John D. Lafferty. Statistical models for text segmentation. Machine Learning*, 34(1-3):177–210, 1999.*

Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In Proceedings of NAACL*, pages 26–33, 2000.*

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

S. O. Ba and J.-M. Odobez. Multi-party focus of attention recognition in meetings from head pose and multimodal contextual cues. In Proc. of ICASSP*, 2008a.*

S. O. Ba, Hayley Hung, and J.-M. Odobez. Visual activity context for focus of attention estimation in dynamic meetings. In ICME*, 2009.*

S. O. Ba and J.-M. Odobez. A Rao-Blackwellized mixed state particle filter for head pose tracking. In Proc. ACM-ICMI-MMMP*, pages 9–16, 2005.*

Silèye O. Ba and Jean-Marc Odobez. Multi-person visual focus of attention from head pose and meeting contextual cues. Technical Report Idiap-RR-47-2008, Idiap Research Institute, 2008b.

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.*

Hayley Hung, Dinesh Babu Jayagopi, Silèye O. Ba, Jean-Marc Odobez, and Daniel Gatica-Perez. Investigating automatic dominance estimation in groups from visual attention and speaking activity. In International Conference on Multi-modal Interfaces*, 2008.*

A. Kalma. Gazing in triads: a powerful signal in floor apportionment. British journal of social psychology*, 31:21–39, 1992.*

K. Otsuka, J. Yamato, Y. Takemae, and H. Murase. Quantifying interpersonal influence in face-to-face conversations based on visual attention patterns. In Proc. ACM CHI Extended Abstract*, Montreal, Apr. 2006.*

D. Jayagopi, H. Hung, C. Yeo, and D. GaticaPerez. Modeling dominance in group conversations from non-verbal activity cues. IEEE Transactions on Audio, Speech and Language Processing, accepted for publication.

J. Griffith. Further considerations concerning the cohesion-performance relation in military settings. Armed Forces & Society*, 34(1):138–147, October 2007.*

L. J. Braaten. Group cohesion: A new multidimensional model. Group*, 15(1):39–55, 1991.*

A.V. Carron, L.R. Brawley, and W.N. Widmeyer. Advances in sport and exercise psychology measurement*, chapter The measurement of cohesiveness in sport groups, pages 213–226. Fitness Information Technology. Morgantown, 1998.*

K.L. Gammage, A.V. Carron, and P.A. Estabrooks. Team cohesion and individual productivity: The influence of the norm for productivity and identifiability of individual effort. Small Group Research*, 32(1):3–18, February 2001.*

A.V. Carron and L.R. Brawley. Cohesion: Conceptual measurement issues. Small Group Research*, 31(1):89–105, February 2000.*

G. L. Siebold. The evolution of the measurement of cohesion. Military Psychology*, 11 (1):5–26, 1999.*

J Dines, J Vepa, and Thomas Hain. The segmentation of multi-channel meeting recordings for automatic speech recognition. In INTERSPEECH*, 2006.*

D. Tannen. Gender and Discourse*, chapter Interpreting Interruption in Conversation, pages 53–83. Oxford Univesrity Press, 1993.*

Pamela J. Hinds and Diane E. Bailey. Out of sight, out of sync: Understanding conflict in distributed teams. Organization Science*, 14(6):615–632, Nov – Dec 2003.*

M. Poel, R.W. Poppe, and A. Nijholt. Meeting behavior detection in smart environments: Nonverbal cues that help to obtain natural interaction. In J. Cohn, T.S. Huang, M. Pantic, and N. Sebe, editors, Proceedings 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2008)*, pages 1–6, Los Alamitos, September 2008. IEEE Computer Society Press. URL* `http://doc.utwente.nl/64947/`*.*

H. J. A. op den Akker, D. H. W. Hofs, G. H. W. Hondorp, H. op den Akker, J. Zwiers, and A. Nijholt. Supporting engagement and floor control in hybrid meetings. In A. Esposito and R. Vich, editors, Cross-Modal Analysis of Speech, Gestures, Gaze and Facial Expressions, Prague*, volume 5641 of* Lecture Notes in Computer Science*, pages 276–290, Berlin, July 2009. Springer Verlag.*

Taemie Kim, Agnes Chang, Lindsey Holland, and Alex (Sandy) Pentland. Meeting mediator: enhancing group collaboration with sociometric feedback. In CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pages 3183–3188, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-X. doi: http://doi.acm.org/10.1145/1358628.1358828.*

K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich, , and J. Yamato. A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization. In Proc. Int. Conf. on Multimodal Interfaces (ICMI'08)*, Chania, Greece, October 2008. ACM.*

Joan Morris DiMicco, Anna Pandolfo, and Walter Bender. Influencing group participation with a shared display. In CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 614–623, New York, NY, USA, 2004. ACM. ISBN 1-58113-810-5. doi: http://doi.acm.org/10.1145/1031607.1031713.*

Brid O'Conaill, Steve Whittaker, and Sylvia Wilbur. Conversations over video conferences: an evaluation of the spoken aspects of video-mediated communication. Human-Computer Interaction*, 8(4):389–428, 1993. ISSN 0737-0024. doi: http://dx.doi.org/10.1207/s15327051hci0804_4.*

Taemie Kim and Alex (Sandy) Pentland. Understanding effects of feedback on group collaboration. Association for the Advancement of Artificial Intelligence*, pages 25–30, 2009.*

D. Gatica-Perez. Automatic nonverbal analysis of social interaction in small groups: a review. Image and Vision Computing, Special Issue on Human Naturalistic Behavior.

D. Jayagopi, B. Raducanu, and D. Gatica-Perez. Characterizing conversational group dynamics using nonverbal behavior. In Proc. IEEE Int. Conf. on Multimedia (ICME)*, New York, June 2009. IEEE.*

J. Hall et al. Nonverbal behavior and the vertical dimension of social relations: A meta-analysis. Psychological bulletin*, 131(6):898–924, 2005.*

M. Lincoln et al. The amida data recording environment. In Proc. Workshop on Machine Learning for Multimodal Interaction (MLMI)*, 2007.*

C. M. Bishop. Pattern Recognition and Machine Learning*. Springer, 2006.*

AMIDA Deliverable 5.4. Deliverable d5.4: Wp5 work in year 2. `http://www.amidaproject.org/`*, October 15 2008.*

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. In Asunción Gómez-Pérez and V. Richard Benjamins, editors, Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. 13th International Conference, EKAW 2002*, pages 166–181, Siguenza, Spain, October 1–4 2002. Springer.*

Aldo Gangemi and Peter Mika. Understanding the Semantic Web through Descriptions and Situations. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, LNCS, Catania, Italy, November 3–7 2003. Springer Berlin/Heidelberg.*

Ralf Engel. Spin: A semantic parser for spoken dialog systems. In Proceedings of the Fifth Slovenian And First International Language Technology Conference (IS-LTC 2006)*, Ljubljana, Slovenia, 2006.*

Nataša Jovanović. To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*. PhD thesis, University of Twente, April 2007.*

Ralf Engel and Daniel Sonntag. Text generation in the smartweb multimodal dialogue system. In KI 2007: Advances in Artificial Intelligence*, volume 4667 of* Lecture Notes in Computer Science*, pages 448–451. Springer Berlin / Heidelberg, August 26 2007.*

M. Sanderson. Word sense disambiguation and information retrieval*. PhD thesis, Department of Computing Science, University of Glasgow, 1996.*

Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 593–596, 2005.*

D. C. van Leijenhorst and Th. P. van der Weide. A formal derivation of heaps' law. Inf. Sci. Inf. Comput. Sci.*, 170(2-4):263–272, 2005. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2004.03.006.*

S. Raaijmakers, Corne Versloot, and Joost de Wit. A cocktail approach to the VideoCLEF'09 linking task. In Francesca Borri, Alessandro Nardi and Carol Peters (eds.), Working Notes of CLEF 2009*, 2009.*

David Milne and Ian H. Witten. Learning to link with Wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge mining (CIKM'08)*, pages 509–518, New York, NY, USA, 2008. ACM.*

S. F. Adafre and Maarten de Rijke. Finding Similar Sentences across Multiple Languages in Wikipedia. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69, 2006.*

Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part–of–speech tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70, 2000b. URL* `http://nlp.stanford.edu/~manning/papers/emnlp2000.pdf`*.*

Erik Hatcher and Otis Gospodnetic. Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA, 2004. ISBN 1932394281.*

Jaana Kekäläinen and Kalervo Järvelin. Using graded relevance assessments in IR evaluation. J. Am. Soc. Inf. Sci. Technol.*, 53(13):1120–1129, 2002. ISSN 1532-2882.*

S. Sabri and B. Prasada. Video conferencing systems. Proceedings of the IEEE*, 73(4): 671 – 688, 1985.*

P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with ferret. In S. Renals and S. Bengio, editors, Proceedings of the 1st Joint Workshop on MLMI*. Springer Verlag, 2004.*

Adam Herout, Radek Kubicek, Pavel Zak, and Pavel Zemcik. Automatic video editing for multimodal meetings. In Proceedings of International Conference on Computer Vision and Graphics*, volume 0, Heidelberg, De, 2008.*

GraphAtFit Youtube channel. Automatic video editing demo, 2009. URL `http://www.youtube.com/watch?v=Kwr1QIBCWRA`.

S. Sumec. Multi camera automatic video editing. In Proceedings of the ICCVG*, pages 935–945. Kluwer Verlag, 2004.*

M. Al-Hames, B. Hörnler, C. Scheuermann, and G. Rigoll. Using audio, visual, and lexical features in a multi-modal virtual meeting director. In Proceedings of the 3rd Joint Workshop on MLMI*. Springer Verlag, 2006.*

M. Al-Hames, B. Hörnler, R. Müller, J. Schenk, and G. Rigoll. Automatic multi-modal meeting camera selection for video-conferences and meeting browsing. In Proceedings of the 8th ICME*, 2007.*

L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE*, 77(2):257–285, 1989.*

Z. Fang, Z. Guoliang, and S. Zhanjiang. Comparison of different implementations of MFCC. Journal of Computer Science and Technology*, 16(6):582–589, 2001.*

F. Wallhoff, M. Zobl, and G. Rigoll. Action segmentation and recognition in meeting room scenarios. In Proceedings of the 11th ICIP*, 2004.*

M. Al-Hames et al. Multimodal integration for meeting group action segmentation and recognition. In Proceedings of the 2nd Joint Workshop on MLMI*, 2006.*

S. Reiter, B. Schuller, and G. Rigoll. Hidden conditional random fields for meeting segmentation. In Proceedings of the 8th 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 PETS-ICVS*, pages 32–36, 2003.*