Agreement Detection in Multiparty Conversation

Sebastian Germesin
DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
sebastian.germesin@dfki.de

Theresa Wilson
School of Informatics
University of Edinburgh
Edinburgh, U.K. EH8 9AB
twilson@inf.ed.ac.uk

ABSTRACT

This paper presents a system for the automatic detection of agreements in multi-party conversations. We investigate various types of features that are useful for identifying agreements, including lexical, prosodic, and structural features. This system is implemented using supervised machine learning techniques and yields competitive results: Accuracy of *98.1%* and a kappa value of *0.4*. We also begin to explore the novel task of detecting the addressee of agreements (which speaker is being agreed with). Our system for this task achieves an accuracy of *80.3%*, a 56% improvement over the baseline.

General Terms

MEASUREMENT, PERFORMANCE, EXPERIMENTATION

Keywords

agreement detection, multi-party conversation

Categories and Subject Descriptors

I.2 [**Natural Language Processing**]: Discourse; I.2 [**Natural Language Processing**]: Text Analysis

1. INTRODUCTION

Over the past several years, there has been a growing interest in extracting and summarising information from meetings. One type of information of particular importance to meeting summaries is information about when someone agrees (or disagrees) with someone else. Such information would be especially important for summaries that focus on decisions, or to assist in tasks such as auditing of past decisions.

This paper describes a system for the automatic detection of agreements[1] and the speaker targets of those agreements in multi-party conversations. Agreement annotations in the AMI corpus [16] are used to develop automatic systems, employing a combination of high-precision rules and machine learning classifiers. The automatic systems exploit a wide variety of features, including lexical, prosodic, and structural features. To the best of our knowledge, this is the first work to investigate the detection of the speaker targets of agreements.

[1]We will use the term agreement for both, agreements and disagreements unless distinction is necessary.

ICMI-MLMI'09, November 2–4, 2009, Cambridge, MA, USA.

2. PREVIOUS WORK

Previous work in the field of detecting agreements has been performed by Hillard et al. [8], Galley et al. [5] and Hahn et al. [7]. In all of these works, they used spurt-level agreement annotations from the ICSI corpus [9]. In [8] the authors take an unsupervised machine learning approach, although the spurt segments they use were adjusted by humans annotators. They defined four categories for classification: *positive, negative, backchannel* and *other*, where single-word spurts from the positive class were annotated as backchannels "because of the trivial nature of their detection and because they may reflect encouragement for the speaker to continue more than actual agreement." They addressed the problem of the skewness of the class distribution (9%, 6%, 23%, 62%) by oversampling instances of the smaller classes. Galley et al. based their work on a slightly different set of the same data with almost the same class distribution and used Bayesian Networks for the detection of agreements. As part of their study, they "first performed several empirical analyses in order to determine to what extent contextual information helps in discrimination between agreements." They used adjacency pair information to determine the structure of their conditional markov model, outperforming the results of Hillard et al. Using a semi-supervised contrast classifier on the same data, Hahn et al. reached a competitive accuracy on hand-transcribed data which was trained on a lexical-only feature set.

3. DATA

The data we used for our experiments is annotated data from the AMI meeting corpus [2]. This corpus contains *scenario* meetings, which last around 30 minutes. In these meetings, four participants are given the task of designing a remote control. Each of the participants plays one of the given specific roles: Project Manager (PM), Industrial Designer (ID), Marketing Expert (ME) and User Interface Designer (UI). These meetings were recorded for both audio and video, transcribed, and enriched with various types of annotations. Although the meetings are somewhat artifi-

cial, meaning that the participants playing a role, the conversations in the meetings do reflect natural, human-human interaction.

Wilson presents in [16] an annotation guide for marking transcriptions of multi-party meetings with information about subjective content. In a first version, she distinguished six different types of subjectivity: *positive, negative, positive and negative, uncertainty* and *other*. In a later adaption, she splits the first class into two separate classes, *positive* and *agreement*. Similarly, the second class is broken into *negative* and *disagreement* classes. Further enrichments, like the target of the agreements, whether the speaker agrees for the whole group or just for himself or what the agreement is about (e.g., a task, the meeting itself, ...), are also included in the annotations. Agreements, in Wilson's annotations, are those that "contain a private state" of the speaker, including things like concession, where an agreement is reluctantly or grudgingly given.

The agreement annotations from the AMI corpus have two advantages over the ICSI data. The first is that 20 meetings have been annotated, which is nearly three times the data from the previous publications. Secondly, the annotations are more detailed and include the targets of agreements.

3.1 Agreement / Disagreement Annotations

In the 20 meetings of the AMI corpus that have been annotated with subjectivity information, there are a total of 636 agreements and 70 disagreements. Using a word-based measurement, this means that 1.59% of all words in the meetings are contained in agreements, and only 0.35% are are contained in disagreements, leaving 98.01% as the remaining. 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% for evaluation. Table 1 shows the division of the meetings between the two sets. The training data is used in the analysis below, as well as 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 1: Training and evaluation part of the data

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 the Industrial Designer, whereas the *target* of the agreement, what the Industrial Designer is agreeing with, is underlined. We refer to the speaker of the target utterance as the *target speaker*.

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

Looking at the length of agreements and disagreements, it is interesting that the average length (in terms of words) of agreements is 2.9 words, whereas disagreements, on average, are more than twice as long (6.3 words). Galley et al. [5] also report on this fact and hypothesize that "speakers need to elaborate more on the reasons and circumstances of their disagreements than their agreements." Looking at the length of the targets, we observe the opposite effect: an average of 9.7 words for the target of agreements and and average of 7.7 words for the targets of disagreements. By their nature, targets are uttered by *another speaker* and should lie *in the near past* of the agreements. To verify this, we measured the time from the first target word to the first agreeing word. This ranges from 0.1 seconds to 47.7 seconds, with an average of 3.5 seconds and a standard deviation of 3.6 seconds. Assuming a normal distribution, we can say that 95% of all targets lie within a durational window of 10.7 seconds.

3.2 Segmentation

One open question is what unit of segmentation is most appropriate when classifying agreements. Previous work classified spurt segments [8, 5, 7]. Spurts are identified automatically using pauses greater than one-half second between words [13]. Because they are very quickly and easily computed, they are handy to use when other types of segmentation are not available. However, spurts also are more likely to split an utterance in the middle of a speaker's thought, which is not desirable for a task such as ours. To avoid this, the authors of [8] supported their segmentation with "human adjustments."

Another possible type of segmentation to use is dialog act segments. Akker et al. [12] defines dialog acts as "a sequence of subsequent words from a single speaker that form a single statement, an intention or an expression." Dialog act segments inherently seem to be closer to the units that we want to identify as agreements and disagreements; however, they have the disadvantage of being more difficult to compute automatically.

Table 2 shows a comparison between the two types of segments in the AMI corpus. In the table, the average segment length is given for manually annotated dialog segments, automatically recognized dialog act segments, and spurts, for both manual transcription and ASR. The automatic dialog act segmentation was performed using the classifier from [12]. We can see that, although automatic dialog act segmentation performs very well on ASR data, it still produces segments that are twice as long as the manual annotations on manual transcription. Spurt are also typically much longer than the manual dialog acts.

The longer spurt segments become problematic once we want to align the agreement annotations to the segments to be classified. For the agreement annotations, words, phrases, and individual statements were marked. If a long segment contains many more words than actual agreeing words, this will lead to much noisier, harder-to-classify data. Because of this, we chose to use dialog act segments as the basis for our work. For experiments using manual transcriptions, such as those presented in this paper, we use the manual dialog act segments. For future experiments using ASR transcriptions, automatic dialog act segments will be used.

In our (training) data, we count a total of 15,148 dialog act segments. We assigned agreement labels to the

words	segm.	μ (length)	σ (length)	$\mu + 2\sigma$
manual transcript	man.	6.2	6.3	18.8
	aut.	11.0	18.0	47.0
	spurt	11.1	19.5	50.1
ASR output	man.	6.7	6.6	19.9
	aut.	6.3	6.5	19.3
	spurt	7.3	9.0	25.3

Table 2: Comparison of manual vs. automatic DA segmentation with spurt segmentation

segments based on overlap with the manual annotations, which yields 549 agreement segments and 66 disagreement segments. This is a distribution of 3.6% agreements, 0.4% disagreements, and 96.0% remaining segments. Applying the same measurement for the duration between agreements and their target, we have an average of 2.5 segments between them with a standard deviation of 2.3 segments. We use this information later in the process of contextual feature development.

4. FEATURES FOR LEARNING

In this section, we describe the various features that were available in our data for use in agreement detection.

4.1 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[2] as presented in [14] 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. [8], 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 3 shows the top keywords according to their effectiveness ratio for both agreements 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 3: Keywords for agreements/disagreements

The positive and negative polarity words are derived from the MPQA subjectivity lexicon [17].

4.2 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.

4.3 Dialog 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 recognizing agreements. We use the manually annotated dialog act labels from the AMI corpus, which contain a total of 15 types of labels.[3] Table 4 presents 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 4: Distribution of DA labels for agreements and disagreements

Table 5 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 knowledge 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 5: Distribution of DA labels for targets

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

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

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

Galley et al. [5] introduced the use of adjacency pairs for agreement detection. An adjacency pair is a unit of conversation that contains two functionally related dialog acts each by a different speaker. In the AMI corpus, we found that 62.7% of (dis)agreements and their targets match in structure to their counterpart in the adjacency pair annotations. For our experiments, we do not use adjacency pairs directly for agreement detection. Instead, we use them in detecting the target of agreements.

5. AUTOMATIC RECOGNITION

Figure 1 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 target detection, in which we determine who the agreement or disagreement actually was directed toward.

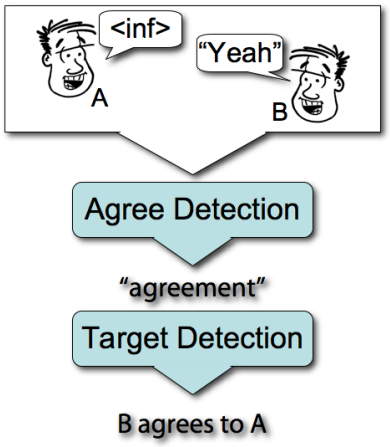

Figure 1: Sketched design of detection system

5.1 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. We hypothesised that this prior knowledge would prove helpful to the machine learners, as the combination of high-precision rules and machine learning has worked quite well for other tasks, such as disfluency detection [6, 11]. We are aware that hand-written rules in a classification approach risk overfitting towards the training data. To address this problem, we focussed on identifying rules for the task that would be as general as possible.

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 preceding 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 $tag(s) = unclassified$.

5.2 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*. Each approach, including its structure and the features used, is described below.

5.2.1 Features

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-dependencies 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. A selection of these features is listed in Table 6, where the index i is the relative position of the segment to the current one.

durationOfSegment$_i$
startsWithYeah$_i$
numberOfWord'NO'Rel$_i$
DALabelOfSegmentOfOtherSpeaker$_i$
DALabelOfSegmentOfSameSpeaker$_i$
DALabelOfSegment$_i$
...

Table 6: Interdependency features for the DT

5.2.2 *Decision Tree*

The first learning method that we applied to agreement detection was decision tree classification. The decision tree learner we used was the C4.5 implementation from the WEKA toolkit[4] - namely J4.8. Although decision trees find skewed distributions problematic, they also provide a model that is informative and can provide insights to the problem. Figure 2 shows the top three levels of the learned decision tree. Looking at the nodes in these top levels, it is interesting to note that these decisions only rely on lexical information (like the starting word of a segment) or dialog act labels. This supports the results of previous studies where lexical features were also identified as the most important features.

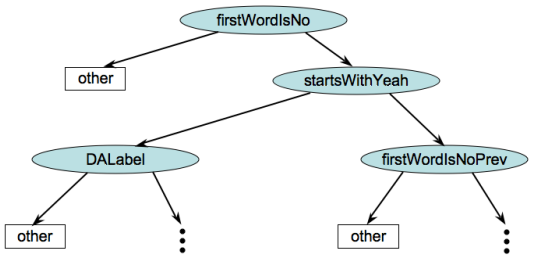

Figure 2: Structure of the trained DT

5.2.3 *Conditional Random Fields*

As the detection of agreements and disagreements would seem to greatly rely on inter-dependencies between different speaker utterances, we also wanted to explore classifiers that could directly model these dependencies. Thus, we use Conditional Random Fields (CRFs) [10] as a second machine learning approach for agreement detection. For our experiments, we used the CRF that is included in the NER toolkit from the Stanford NLP group [4].[5] This package comes with a fully connected CRF, which was easily changed for our purposes. Figure 3 shows the CRF model that we used, where time is displayed from left to right, and the rightmost node represents the segment to be labeled. We found that a window of three segments, each connected to the current segment, using Viterbi search, is the best configuration for the agreement detection in our environment.

5.3 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 3.1.

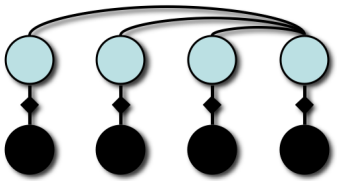

Figure 3: Structure of the used CRF

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

Table 7 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. Our algorithm for identifying target speakers is given below.

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

Table 7: Addressee distribution in the AMI corpus

Algorithm 1 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)
        return
      end if
    end for
    {Fallback}
    if s has no addressee yet then
      s.addressee = getIndex(getLastSpeaker())
    end if
  end if
end for
```

6. EVALUATION

This section presents results for agreement detection and target speaker detection on the held-out evaluation data. We provide detailed information about the performance of each subsystem individually in the order they appear in the detection system, as shown in Figure 1.

[4]http://www.cs.waikato.ac.nz/~ml/weka/
[5]http://nlp.stanford.edu/software/CRF-NER.shtml

6.1 High-Precision Rules

Table 8 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. We can see that all of the rules perform very well on the evaluation set, though explicitly developed on the training set.

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 8: Evaluation of High-Precision Rules

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

6.2 (Dis-)Agreement Detection

The performance of the (dis-)agreement detection for each approach is given in Table 9. 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 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. Both systems perform very fast on the data, resulting in a real-time factor way below zero.

Unfortunately, neither of the presented 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.

6.3 Target Detection

Table 10 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. Recall that the baseline is assuming the last active speaker to be the target speaker for any case. 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 10: Evaluation of Target Detection

7. CONCLUSIONS AND FUTURE WORK

This paper presents our work on the detection of agreements and disagreements in multi-party conversations. We utilize a wide variety of features, including lexical, prosodic, and structural features, and experiment with two different types of machine learning for this task, decision trees and CRFs. Both systems achieved comparable results in terms of accuracy (98.1%) and kappa (0.4), but there were clear differences in terms of recall and precision. The CRF classifier achieved much higher precision, while the decision tree classifier had the higher recall.

One novel aspect of this work is the detection of the speaker who is the target of the agreement. For this task, we found that adjacency pair information is likely very important, but it is not the whole picture. Incorporating adjacency pair information gives a 56% improvement over the baseline. Our current system is a simple rule-based system; in the future, we plan to investigate machine learning approaches to the problem.

For the work in this paper, the dialog act segments that were classified and many of the features were based on manual annotations. Moving to a completely automatic setup for both dialog act segments and features is an important aspect of our ongoing work. This includes working with ASR, rather than manual transcriptions, with the end goal of developing an on-line system for agreement detection.

When we examined the errors the classifier made, we found that it focuses the detection mainly on one-word agreements (e.g., 'yeah'). Therefore, we believe that an approach where we have two classifiers, one trained for the detection of short agreements and one trained for the detection of longer agreements will be an important avenue to pursue. We also plan to investigate the use of separate disagreement detectors.

An obvious challenge to agreement detection is the class imbalance, in particular for detecting disagreements. Hillard et al. [8] addressed the problem by oversampling. However, a variety of methods have been proposed for oversampling, which may be more or less appropriate depending on the data [1]. Other methods for dealing with skewed data have also been proposed, such as specialized feature selection for the different classes and one-class learning [3]. How best to mitigate the effects of the class imbalance for this data is clearly an open question.

Lastly, incorporating new features from other modalities (e.g., nodding detected from the video signal [15]) will be an interesting and potentially very important line of future research. We suspect that such visual cues will be particularly valuable for detecting agreements in short utterances, which are easily confused with backchannels.

	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 9: Segment-based evaluation of (Dis-)Agreement Detection

8. ACKNOWLEDGMENT

This work is supported by the European IST Programme Project [FP6-0033812] (AMIDA), Publication ID - AMIDA-26, and the European 7th Framework Programme [FP7/2007-2013] under grant agreement 231287 (SSPNet). This paper only reflects the authors views and funding agencies are not liable for any use that may be made of the information contained herein.

9. REFERENCES

[1] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations*, 6(1):20–29, 2004.

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

[3] N. V. Chawla, N. Japkowicz, and A. Kolcz. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations*, 6(1), 2004.

[4] J. Finkel, T. Grenager, and C. Manning. Annotation and Analysis of Emotionally Relevant Behavior in the ISL Meeting Corpus. In *Proceedings of ACL*, 2005.

[5] M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. Identifying Agreement and Disagreement in Conversational Speech: Use of Bayesian Networks to Model Pragmatic Dependencies. In *Proceedings of ACL*, 2004.

[6] S. Germesin, T. Becker, and P. Poller. Domain-specific Classification Methods for Disfluency Detection. In *Proceedings of Interspeech*, 2008.

[7] S. Hahn, R. Ladner, and M. Ostendorf. Agreement/Disagreement Classification: Exploiting Unlabeled Data Using Contrast Classifiers. In *Proceedings of HLT/NAACL*, 2006.

[8] D. Hillard, M. Ostendorf, and E. Shriberg. Detection of Agreement vs. Disagreement in Meetings: Training with Unlabeled Data. In *Proceedings of HLT/NAACL*, 2003.

[9] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, and E. Shriberg. The ICSI Meeting Corpus. In *Proceedings of ICASSP*, 2003.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, 2001.

[11] M. Mieskes and M. Strube. A Three-Stage Disfluency Classifier for Multi-Party Dialogues. In *Proceedings of LREC*, 2008.

[12] H. op den Akker and C. Schulz. Exploring Features and Classifiers for Dialogue Act Segmentation. In *Proceedings of MLMI*, 2008.

[13] E. Shriberg, A. Stolcke, and D. Baron. Observations on Overlap: Findings and Implications for Automatic Processing of Multi-Party Conversation. In *Proceedings of Eurospeech*, 2001.

[14] K. Toutanova and C. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of EMNLP/VLC-2000*, 2000.

[15] F. Wallhoff, M. Zobl, and G. Rigoll. Action Segmentation and Recognition in Meeting Room Scenarios. In *Proceedings of ICIP*, pages 2223–2226, 2004.

[16] T. Wilson. Annotating Subjective Content in Meetings. In *Proceedings of LREC*, 2008.

[17] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of HLT-EMNLP*, 2005.