Identifying Agreement and Disagreement in Conversational Speech: Use of Bayesian Networks to Model Pragmatic Dependencies

Michel Galley[*], **Kathleen McKeown**[*], **Julia Hirschberg**[*], and **Elizabeth Shriberg**[†]

[*]Columbia University
Computer Science Department
1214 Amsterdam Avenue
New York, NY 10027, USA
{galley,kathy,julia}@cs.columbia.edu

[†]SRI International
Speech Technology and Research Laboratory
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
ees@speech.sri.com

Abstract

We describe a statistical approach for modeling agreements and disagreements in conversational interaction. Our approach first identifies adjacency pairs using maximum entropy ranking based on a set of lexical, durational, and structural features that look both forward and backward in the discourse. We then classify utterances as agreement or disagreement using these adjacency pairs and features that represent various pragmatic influences of previous agreement or disagreement on the current utterance. Our approach achieves 86.9% accuracy, a 4.9% increase over previous work.

1 Introduction

One of the main features of meetings is the occurrence of agreement and disagreement among participants. Often meetings include long stretches of controversial discussion before some consensus decision is reached. Our ultimate goal is automated summarization of multi-participant meetings and we hypothesize that the ability to automatically identify agreement and disagreement between participants will help us in the summarization task. For example, a summary might resemble minutes of meetings with major decisions reached (consensus) along with highlighted points of the pros and cons for each decision. In this paper, we present a method to automatically classify utterances as agreement, disagreement, or neither.

Previous work in automatic identification of agreement/disagreement (Hillard et al., 2003) demonstrates that this is a feasible task when various textual, durational, and acoustic features are available. We build on their approach and show that we can get an improvement in accuracy when contextual information is taken into account. Our approach first identifies adjacency pairs using maximum entropy ranking based on a set of lexical, durational and structural features that look both forward and backward in the discourse. This allows us to acquire, and subsequently process, knowledge about who speaks to whom. We hypothesize that pragmatic features that center around previous agreement between speakers in the dialog will influence the determination of agreement/disagreement. For example, if a speaker disagrees with another person once in the conversation, is he more likely to disagree with him again? We model context using Bayesian networks that allows capturing of these pragmatic dependencies. Our accuracy for classifying agreements and disagreements is 86.9%, which is a 4.9% improvement over (Hillard et al., 2003).

In the following sections, we begin by describing the annotated corpus that we used for our experiments. We then turn to our work on identifying adjacency pairs. In the section on identification of agreement/disagreement, we describe the contextual features that we model and the implementation of the classifier. We close with a discussion of future work.

2 Corpus

The ICSI Meeting corpus (Janin et al., 2003) is a collection of 75 meetings collected at the International Computer Science Institute (ICSI), one among the growing number of corpora of human-to-human multi-party conversations. These are naturally occurring, regular weekly meetings of various ICSI research teams. Meetings in general run just under an hour each; they have an average of 6.5 participants.

These meetings have been labeled with adjacency pairs (AP), which provide information about speaker interaction. They reflect the structure of conversations as paired utterances such as question-answer and offer-acceptance, and their labeling is used in our work to determine who are the addressees in agreements and disagreements. The annotation of the corpus with adjacency pairs is described in (Shriberg et al., 2004; Dhillon et al., 2004).

Seven of those meetings were segmented into spurts, defined as periods of speech that have no pauses greater than .5 second, and each spurt was labeled with one of the four categories: agreement, disagreement, backchannel, and other.[1] We used spurt segmentation as our unit of analysis instead of sentence segmentation, because our ultimate goal is to build a system that can be fully automated, and in that respect, spurt segmentation is easy to obtain. Backchannels (e.g. "uhhuh" and "okay") were treated as a separate category, since they are generally used by listeners to indicate they are following along, while not necessarily indicating agreement. The proportion of classes is the following: 11.9% are agreements, 6.8% are disagreements, 23.2% are backchannels, and 58.1% are others. Inter-labeler reliability estimated on 500 spurts with 2 labelers was considered quite acceptable, since the kappa coefficient was .63 (Cohen, 1960).

3 Adjacency Pairs

3.1 Overview

Adjacency pairs (AP) are considered fundamental units of conversational organization (Schegloff and Sacks, 1973). Their identification is central to our problem, since we need to know the identity of addressees in agreements and disagreements, and adjacency pairs provide a means of acquiring this knowledge. An adjacency pair is said to consist of two parts (later referred to as A and B) that are ordered, adjacent, and produced by different speakers. The first part makes the second one immediately relevant, as a question does with an answer, or an offer does with an acceptance. Extensive work in conversational analysis uses a less restrictive definition of adjacency pair that does not impose any actual adjacency requirement; this requirement is problematic in many respects (Levinson, 1983). Even when APs are not directly adjacent, the same constraints between pairs and mechanisms for selecting the next speaker remain in place (e.g. the case of embedded question and answer pairs). This relaxation on a strict adjacency requirement is particularly important in interactions of multiple speakers since other speakers have more opportunities to insert utterances between the two elements of the AP construction (e.g. interrupted, abandoned or ignored utterances; backchannels; APs with multiple second elements, e.g. a question followed by answers of multiple speakers).[2]

Information provided by adjacency pairs can be used to identify the target of an agreeing or disagreeing utterance. We define the problem of AP identification as follows: given the second element (B) of an adjacency pair, determine who is the speaker of the first element (A). A quite effective baseline algorithm is to select as speaker of utterance A the most recent speaker before the occurrence of utterance B. This strategy selects the right speaker in 79.8% of the cases in the 50 meetings that were annotated with adjacency pairs. The next subsection describes the machine learning framework used to significantly outperform this already quite effective baseline algorithm.

3.2 Maximum Entropy Ranking

We view the problem as an instance of statistical ranking, a general machine learning paradigm used for example in statistical parsing (Collins, 2000) and question answering (Ravichandran et al., 2003).[3] The problem is to select, given a set of N possible candidates $\{s_1, ..., s_N\}$ (in our case, potential A speakers), the one candidate s_i that maximizes a given conditional probability distribution.

We use maximum entropy modeling (Berger et al., 1996) to directly model the conditional probability $p(s_i|\mathbf{d})$, where each d_i in $\mathbf{d} = (d_1, ..., d_N)$ is an observation associated with the corresponding speaker s_i. d_i is represented here by only one variable for notational ease, but it possibly represents several lexical, durational, structural, and acoustic observations. Given J feature functions $f_j(\mathbf{d}, s_i)$ and J model parameters $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_J)$, the probability of the maximum entropy model is defined as:

$$p_{\boldsymbol{\lambda}}(s_i|\mathbf{d}) = \frac{1}{Z(\mathbf{d})} \exp\left(\sum_{j=1}^{J} \lambda_j f_j(\mathbf{d}, s_i)\right)$$

The only role of the denominator $Z(\mathbf{d})$ is to ensure that $p_{\boldsymbol{\lambda}}$ is a proper probability distribution. It is defined as:

$$Z(\mathbf{d}) = \sum_{i'=1}^{N} \exp\left(\sum_{j=1}^{J} \lambda_j f_j(\mathbf{d}, s_{i'})\right)$$

To find the most probable speaker of part A, we use the following decision rule:

$$\begin{aligned}\hat{s} &= \underset{s_i \in \{s_1, ..., s_N\}}{\arg\max} \left\{ p_{\boldsymbol{\lambda}}(s_i|\mathbf{d}) \right\} \\ &= \underset{s_i \in \{s_1, ..., s_N\}}{\arg\max} \left\{ \exp\left(\sum_{j=1}^{J} \lambda_j f_j(\mathbf{d}, s_i)\right) \right\}\end{aligned}$$

Note that we have also attempted to model the problem as a binary classification problem where

[1] Part of these annotated meetings were provided by the authors of (Hillard et al., 2003).

[2] The percentage of APs labeled in our data that have non-contiguous parts is about 21%.

[3] The approach is generally called re-ranking in cases where candidates are assigned an initial rank beforehand.

each speaker is either classified as speaker A or not, but we abandoned that approach, since it gives much worse performance. This finding is consistent with previous work (Ravichandran et al., 2003) that compares maximum entropy classification and re-ranking on a question answering task.

3.3 Features

We will now describe the features used to train the maximum entropy model mentioned previously. To rank all speakers (aside from the B speaker) and to determine how likely each one is to be the A speaker of the adjacency pair involving speaker B, we use four categories of features: structural, durational, lexical, and dialog act (DA) information. For the remainder of this section, we will interchangeably use A to designate either the *potential* A speaker or the most recent utterance[4] of that speaker, assuming the distinction is generally unambiguous. We use B to designate either the B speaker or the current spurt for which we need to identify a corresponding A part.

The feature sets are listed in Table 1. Structural features encode some helpful information regarding ordering and overlap of spurts. Note that with only the first feature listed in the table, the maximum entropy ranker matches exactly the performance of the baseline algorithm (79.8% accuracy). Regarding lexical features, we used a count-based feature selection algorithm to remove many first-word and last-word features that occur infrequently and that are typically uninformative for the task at hand. Remaining features essentially contained function words, in particular sentence-initial indicators of questions (“where”, “when”, and so on).

Note that all features in Table 1 are “backward-looking”, in the sense that they result from an analysis of context preceding B. For many of them, we built equivalent “forward-looking” features that pertain to the closest utterance of the potential speaker A that follows part B. The motivation for extracting these features is that speaker A is generally expected to react if he or she is addressed, and thus, to take the floor soon after B is produced.

3.4 Results

We used the labeled adjacency pairs of 50 meetings and selected 80% of the pairs for training. To train the maximum entropy ranking model, we used the generalized iterative scaling algorithm (Darroch and Ratcliff, 1972) as implemented in YASMET.[5]

Structural features:

· number of speakers taking the floor between A and B
· number of spurts between A and B
· number of spurts of speaker B between A and B
· do A and B overlap?

Durational features:

· duration of A
· if A and B do not overlap: time separating A and B
· if they do overlap: duration of overlap
· seconds of overlap with any other speaker
· speech rate in A

Lexical features:

· number of words in A
· number of content words in A
· ratio of words of A (respectively B) that are also in B (respectively A)
· ratio of content words of A (respectively B) that are also in B (respectively A)
· number of n-grams present both in A and B (we built 3 features for n ranging from 2 to 4)
· first and last word of A
· number of instances at any position of A of each cue word listed in (Hirschberg and Litman, 1994)
· does A contain the first/last name of speaker B?

Table 1. Speaker ranking features

Feature sets	Accuracy
Baseline	79.80%
Structural	83.97%
Durational	84.71%
Lexical	75.43%
Structural and durational	87.88%
All	89.38%
All (only backward looking)	86.99%
All (Gaussian smoothing, FS)	90.20%

Table 2. Speaker ranking accuracy

Table 2 summarizes the accuracy of our statistical ranker on the test data with different feature sets: the performance is 89.39% when using all feature sets, and reaches 90.2% after applying Gaussian smoothing and using incremental feature selection as described in (Berger et al., 1996) and implemented in the yasmetFS package.[6] Note that restricting ourselves to only backward looking features decreases the performance significantly, as we can see in Table 2.

We also wanted to determine if information about

[4] We build features for both the entire speaker turn of A and the most recent spurt of A.

[5] `http://www.isi.edu/˜och/YASMET.html`

[6] `http://www.isi.edu/˜ravichan/YASMET.html`

dialog acts (DA) helps the ranking task. If we hypothesize that only a limited set of paired DAs (e.g. offer-accept, question-answer, and apology-downplay) can be realized as adjacency pairs, then knowing the DA category of the B part and of all potential A parts should help in finding the most meaningful dialog act tag among all potential A parts; for example, the question-accept pair is admittedly more likely to correspond to an AP than e.g. backchannel-accept. We used the DA annotation that we also had available, and used the DA tag sequence of part A and B as a feature.[7]

When we add the DA feature set, the accuracy reaches 91.34%, which is only slightly better than our 90.20% accuracy, which indicates that lexical, durational, and structural features capture most of the informativeness provided by DAs. This improved accuracy with DA information should of course not be considered as the actual accuracy of our system, since DA information is difficult to acquire automatically (Stolcke et al., 2000).

4 Agreements and Disagreements

4.1 Overview

This section focusses on the use of contextual information, in particular the influence of previous agreements and disagreements and detected adjacency pairs, to improve the classification of agreements and disagreements. We first define the classification problem, then describe non-contextual features, provide some empirical evidence justifying our choice of contextual features, and finally evaluate the classifier.

4.2 Agreement/Disagreement Classification

We need to first introduce some notational conventions and define the classification problem with the agreement/disagreement tagset. In our classification problem, each spurt c_i among the L spurts of a meeting must be assigned a tag $c_i \in \{\textsc{Agree}, \textsc{Disagree}, \textsc{Backchannel}, \textsc{Other}\}$.

To specify the speaker of the spurt (e.g. speaker B), the notation will sometimes be augmented to incorporate speaker information, as with c_i^B, and to designate the addressee of B (e.g. listener A), we will use the notation $c_i^{B \to A}$. For example, $c_i^{B \to A} = \textsc{Agree}$ simply means that B agrees with A in the spurt of index i. This notation makes it obvious that we do not necessarily assume that agreements and disagreements are reflexive relations. We define:

$$\operatorname*{pred}_{Y \to X}\left(c_i^{B \to A}\right)$$

as the tag of the most recent spurt before $c_i^{B \to A}$ that is produced by Y and addresses X. This definition will help our multi-party analyses of agreement and disagreement behaviors.

4.3 Local Features

Many of the local features described in this subsection are similar in spirit to the ones used in the previous work of (Hillard et al., 2003). We did not use acoustic features, since the main purpose of the current work is to explore the use of contextual information.

Table 3 lists the features that were found most helpful at identifying agreements and disagreements. Regarding lexical features, we selected a list of lexical items we believed are instrumental in the expression of agreements and disagreements: agreement markers, e.g. “yes” and “right”, as listed in (Cohen, 2002), general cue phrases, e.g. “but” and “alright” (Hirschberg and Litman, 1994), and adjectives with positive or negative polarity (Hatzivassiloglou and McKeown, 1997). We incorporated a set of durational features that were described in the literature as good predictors of agreements: utterance length distinguishes agreement from disagreement, the latter tending to be longer since the speaker elaborates more on the reasons and circumstances of her disagreement than for an agreement (Cohen, 2002). Duration is also a good predictor of backchannels, since they tend to be quite short. Finally, a fair amount of silence and filled pauses is sometimes an indicator of disagreement, since it is a dispreferred response in most social contexts and can be associated with hesitation (Pomerantz, 1984).

4.4 Contextual Features: An Empirical Study

We first performed several empirical analyses in order to determine to what extent contextual information helps in discriminating between agreement and disagreement. By integrating the interpretation of the pragmatic function of an utterance into a wider context, we aim to detect cases of mismatch between a correct pragmatic interpretation and the surface form of the utterance, e.g. the case of weak or “empty” agreement, which has some properties of downright agreement (lexical items of positive polarity), but which is commonly considered to be a disagreement (Pomerantz, 1984).

While the actual classification problem incorporates four classes, the BACKCHANNEL class is ig-

[7] The annotation of DA is particularly fine-grained with a choice of many optional tags that can be associated with each DA. To deal with this problem, we used various scaled-down versions of the original tagset.

Structural features:

· is the previous/next spurt of the same speaker?
· is the previous/next spurt involving the same B speaker?

Durational features:

· duration of the spurt
· seconds of overlap with any other speaker
· seconds of silence during the spurt
· speech rate in the spurt

Lexical features:

· number of words in the spurt
· number of content words in the spurt
· perplexity of the spurt with respect to four language models, one for each class
· first and last word of the spurt
· number of instances of adjectives with positive polarity (Hatzivassiloglou and McKeown, 1997)
· idem, with adjectives of negative polarity
· number of instances in the spurt of each cue phrase and agreement/disagreement token listed in (Hirschberg and Litman, 1994; Cohen, 2002)

Table 3. Local features for agreement and disagreement classification

nored here to make the empirical study easier to interpret. We assume in that study that accurate AP labeling is available, but for the purpose of building and testing a classifier, we use only automatically extracted adjacency pair information. We tested the validity of four pragmatic assumptions:

1. **previous tag dependency:** a tag c_i is influenced by its predecessor c_{i-1}
2. **same-interactants previous tag dependency:** a tag $c_i^{B \to A}$ is influenced by $\text{pred}_{B\to A}\left(c_i^{B\to A}\right)$, the most recent tag of the same speaker addressing the same listener; for example, it might be reasonable to assume that if speaker B disagrees with A, B is likely to disagree with A in his or her next speech addressing A.
3. **reflexivity:** a tag $c_i^{B\to A}$ is influenced by $\text{pred}_{A\to B}\left(c_i^{B\to A}\right)$; the assumption is that c_i^B is influenced by the polarity (agreement or disagreement) of what A said last to B.
4. **transitivity:** assuming there is a speaker X for which $\text{pred}_{X\to A}\left(\text{pred}_{B\to X}\left(c_i^{B\to A}\right)\right)$ exists, then a tag $c_i^{B\to A}$ is influenced by $\text{pred}_{B\to X}\left(c_i^{B\to A}\right)$ and $\text{pred}_{X\to A}\left(\text{pred}_{B\to X}\left(c_i^{B\to A}\right)\right)$; an example of such an influence is a case where speaker X first agrees with A, then speaker B disagrees with X, from which one could possibly conclude that B is actually in disagreement with A.

Table 4 presents the results of our empirical evaluation of the first three assumptions. For comparison, the distribution of classes is the following: 18.8% are agreements, 10.6% disagreements, and 70.6% other. The dependencies empirically evaluated in the two last columns are non-local; they create dependencies between spurts separated by an arbitrarily long time span. Such long range dependencies are often undesirable, since the influence of one spurt on the other is often weak or too difficult to capture with our model. Hence, we made a Markov assumption by limiting context to an arbitrarily chosen value N. In this analysis subsection and for all classification results presented thereafter, we used a value of $N = 10$.

The table yields some interesting results, showing quite significant variations in class distribution when it is conditioned on various types of contextual information. We can see for example, that the proportion of agreements and disagreements (respectively 18.8% and 10.6%) changes to 13.9% and 20.9% respectively when we restrict the counts to spurts that are preceded by a DISAGREE. Similarly, that distribution changes to 21.3% and 7.3% when the previous tag is an AGREE. The variable is even more noticeable between probabilities $p(c_i)$ and $p(c_i^{B\to A} | \, \text{pred}_{B\to A}(c_i^{B\to A}))$. In 26.1% of the cases where a given speaker B disagrees with A, he or she will continue to disagree in the next exchange involving the same speaker and the same listener. Similarly with the same probability distribution, a tendency to agree is confirmed in 25% of the cases. The results in the last column are quite different from the two preceding ones. While agreements in response to agreements ($p(\text{AGREE}|\text{AGREE}) = .175$) are slightly less probable than agreements without conditioning on any previous tag ($p(\text{AGREE}) = .188$), the probability of an agreement produced in response to a disagreement is quite high (with 23.4%), even higher than the proportion of agreements in the entire data (18.8%). This last result would arguably be quite different with more quarrelsome meeting participants.

Table 5 represents results concerning the fourth pragmatic assumption. While none of the results characterize any strong conditioning of c_k by c_i and c_j, we can nevertheless notice some interesting phenomena. For example, there is a tendency for agreements to be transitive, i.e. if X agrees with A and B agrees with X within a limited segment of speech, then agreement between B and A is confirmed in 22.5% of the cases, while the probability of the agreement class is only 18.8%. The only slightly surprising result appears in the last column of the table, from which we cannot conclude that disagreement with a disagreement is equivalent to agreement. This might be explained by the fact that these sequences of agreement and disagreement do not necessarily concern the same propositional content.

The probability distributions presented here are admittedly dependent on the meeting genre and particularly speaker personalities. Nonetheless, we believe this model can as well be used to capture salient interactional patterns specific to meetings with different social dynamics.

We will next discuss our choice of a statistical model to classify sequence data that can deal with non-local label dependencies, such as the ones tested in our empirical study.

4.5 Sequence Classification with Maximum Entropy Models

Extensive research has targeted the problem of labeling sequence information to solve a variety of problems in natural language processing. Hidden Markov models (HMM) are widely used and considerably well understood models for sequence labeling. Their drawback is that, as most generative models, they are generally computed to maximize the joint likelihood of the training data. In order to define a probability distribution over the sequences of observation and labels, it is necessary to enumerate all possible sequences of observations. Such enumeration is generally prohibitive when the model incorporates many interacting features and long-range dependencies (the reader can find a discussion of the problem in (McCallum et al., 2000)).

Conditional models address these concerns. Conditional Markov models (CMM) (Ratnaparkhi, 1996; Klein and Manning, 2002) have been successfully used in sequence labeling tasks incorporating rich feature sets. In a left-to-right CMM as shown in Figure 1(a), the probability of a sequence of L tags $\mathbf{c} = (c_1, ..., c_L)$ is decomposed as:

$$p(\mathbf{c}|\mathbf{d}) = p(c_0) \prod_{i=1}^{L} p(c_i|c_{i-1}, d_i)$$

$\mathbf{d} = (d_1, ..., d_L)$ is the vector of observations and each i is the index of a spurt. The probability distribution $p(c_i|c_{i-1}, d_i)$ associated with each state of the Markov chain only depends on the preceding tag c_{i-1} and the local observation d_i. However, in order to incorporate more than one label dependency and, in particular, to take into account the four pragmatic contextual dependencies discussed in the previous subsection, we must augment the structure of our model to obtain a more general one. Such a model is shown in Figure 1(b), a Bayesian network model that is well-understood and that has precisely defined semantics.

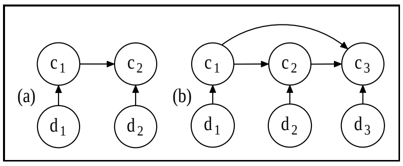

Figure 1. (a) Left-to-right CMM. (b) More complex Bayesian network. Assuming for example that $c_1 = c_1^{B\to A}$ and $c_3 = c_3^{B\to A}$, there is then a direct dependency between c_1 and c_3, and the probability model becomes $p(\mathbf{c}|\mathbf{d}) = p(c_1|d_1)p(c_2|c_1, d_2)p(c_3|c_2, c_1, d_3)$. This is a simplifying example; in practice, each label is dependent on a fixed number of other labels.

To this Bayesian network representation, we apply maximum entropy modeling to define a probability distribution at each node (c_i) dependent on the observation variable d_i and the five contextual tags used in the four pragmatic dependencies.[8] For notational simplicity, the contextual tags representing these pragmatic dependencies are represented here as a vector $\mathbf{p}$ (c_{i-1}, $\text{pred}_{B\to A}(c_i^{B\to A})$, and so on).

Given J feature functions $f_j(\mathbf{p}, d_i, c_i)$ (both local and contextual, like previous tag features) and J model parameters $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_J)$, the probability of the model is defined as:

$$p_{\boldsymbol{\lambda}}(c_i|\mathbf{p}, d_i) = \frac{1}{Z(\mathbf{p}, d_i)} \exp\left(\sum_{j=1}^{J} \lambda_j f_j(\mathbf{p}, d_i, c_i)\right)$$

Again, the only role of the denominator $Z(\mathbf{d})$ is to ensure that $p_{\boldsymbol{\lambda}}$ sums to 1, and need not be computed when searching for the most probable tags. Note that in our case, the structure of the Bayesian network is known and need not be inferred, since AP identification is performed before the actual agreement and disagreement classification. Since tag sequences are known during training, the inference of a model for sequence labels is no more difficult than inferring a model in a non-sequential case.

We compute the most probable sequence by performing a left-to-right decoding using a beam search. The algorithm is exactly the same as the one described in (Ratnaparkhi, 1996) to find the most probable part-of-speech sequence. We used a large beam of size N=100, which is not computationally prohibitive, since the tagset contains only four ele-

[8] The transitivity dependency is conditioned on two tags, while all others on only one. These five contextual tags are defaulted to OTHER when dependency spans exceed the threshold of $N = 10$.

	$p(c_i\|c_{i-1})$	$p(c_i^{B\to A}\|\text{pred}_{B\to A}(c_i^{B\to A}))$	$p(c_i^{B\to A}\|\text{pred}_{A\to B}(c_i^{B\to A}))$
$p(\text{AGREE}\|\text{AGREE})$	**.213**	**.250**	.175
$p(\text{OTHER}\|\text{AGREE})$	.713	.643	.737
$p(\text{DISAGREE}\|\text{AGREE})$	.073	.107	.088
$p(\text{AGREE}\|\text{OTHER})$	.187	.115	.177
$p(\text{OTHER}\|\text{OTHER})$	.714	.784	.710
$p(\text{DISAGREE}\|\text{OTHER})$	.098	.100	.113
$p(\text{AGREE}\|\text{DISAGREE})$	.139	.087	**.234**
$p(\text{OTHER}\|\text{DISAGREE})$	.651	.652	.638
$p(\text{DISAGREE}\|\text{DISAGREE})$	**.209**	**.261**	.128

Table 4. Contextual dependencies (previous tag, same-interactants previous tag, and reflexivity)

	$p(c_k^{B\to A}\|c_i, c_j)$, where $c_j = \text{pred}_{B\to X}(c_k^{B\to A})$ and $c_i = \text{pred}_{X\to A}(c_j)$			
	$c_i = \text{AGREE}$	$c_i = \text{AGREE}$	$c_i = \text{DISAGREE}$	$c_i = \text{DISAGREE}$
	$c_j = \text{AGREE}$	$c_j = \text{DISAGREE}$	$c_j = \text{AGREE}$	$c_j = \text{DISAGREE}$
$p(\text{AGREE}\|c_i, c_j)$	**.225**	.147	.131	.152
$p(\text{OTHER}\|c_i, c_j)$	.658	.677	.683	.668
$p(\text{DISAGREE}\|c_i, c_j)$	.117	**.177**	**.186**	**.180**

Table 5. Contextual dependencies (transitivity)

ments. Note however that this algorithm can lead to search errors. An alternative would be to use a variant of the Viterbi algorithm, which was successfully used in (McCallum et al., 2000) to decode the most probable sequence in a CMM.

4.6 Results

We had 8135 spurts available for training and testing, and performed two sets of experiments to evaluate the performance of our system. The tools used to perform the training are the same as those described in section 3.4. In the first set of experiments, we reproduced the experimental setting of (Hillard et al., 2003), a three-way classification (BACKCHANNEL and OTHER are merged) using hand-labeled data of a single meeting as a test set and the remaining data as training material; for this experiment, we used the same training set as (Hillard et al., 2003). Performance is reported in Table 6. In the second set of experiments, we aimed at reducing the expected variance of our experimental results and performed N-fold cross-validation in a four-way classification task, at each step retaining the hand-labeled data of a meeting for testing and the rest of the data for training. Table 7 summarizes the performance of our classifier with the different feature sets in this classification task, distinguishing the case where the four label-dependency pragmatic features are available during decoding from the case where they are not.

First, the analysis of our results shows that with our three local feature sets only, we obtain substantially better results than (Hillard et al., 2003). This might be due to some additional features the latter work didn't exploit (e.g. structural features and adjective polarity), and to the fact that the learning algorithm used in our experiments might be more accurate than decision trees in the given task. Second, the table corroborates the findings of (Hillard et al., 2003) that lexical information make the most helpful local features. Finally, we observe that by incorporating label-dependency features representing pragmatic influences, we further improve the performance (about 1% in Table 7). This seems to indicate that modeling label dependencies in our classification problem is useful.

Feature sets	Accuracy
(Hillard et al., 2003)	82%
Lexical	84.95%
Structural and durational	71.23%
All (no label dependencies)	85.62%
All (with label dependencies)	86.92%

Table 6. 3-way classification accuracy

Feature sets	Label dep.	No label dep.
Lexical	83.54%	82.62%
Structural, durational	62.10%	58.86%
All	84.07%	83.11%

Table 7. 4-way classification accuracy

5 Conclusion

We have shown how identification of adjacency pairs can help in designing features representing pragmatic dependencies between agreement and disagreement labels. These features are shown to be informative and to help the classification task, yielding a substantial improvement (1.3% to reach a 86.9% accuracy in three-way classification).

We also believe that the present work may be useful in other computational pragmatic research focusing on multi-party dialogs, such as dialog act (DA) classification. Most previous work in that area is limited to interaction between two speakers (e.g. Switchboard, (Stolcke et al., 2000)). When more than two speakers are involved, the question of who is the addressee of an utterance is crucial, since it generally determines what DAs are relevant after the addressee's last utterance. So, knowledge about adjacency pairs is likely to help DA classification.

In future work, we plan to extend our inference process to treat speaker ranking (i.e. AP identification) and agreement/disagreement classification as a single, joint inference problem. Contextual information about agreements and disagreements can also provide useful cues regarding who is the addressee of a given utterance. We also plan to incorporate acoustic features to increase the robustness of our procedure in the case where only speech recognition output is available.

Acknowledgments

We are grateful to Mari Ostendorf and Dustin Hillard for providing us with their agreement and disagreement labeled data.

This material is based on research supported by the National Science Foundation under Grant No. IIS-012196. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

A. Berger, S. Della Pietra, and V Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological measurements*, 20:37–46.

S. Cohen. 2002. A computerized scale for monitoring levels of agreement during a conversation. In *Proc. of the 26th Penn Linguistics Colloquium*.

M. Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. 17th International Conf. on Machine Learning*, pages 175–182.

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

R. Dhillon, S. Bhagat, H. Carvey, and E. Shriberg. 2004. Meeting recorder project: Dialog act labeling guide. Technical Report TR-04-002, ICSI.

V. Hatzivassiloglou and K. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proc. of ACL*.

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

J. Hirschberg and D. Litman. 1994. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI meeting corpus. In *Proc. of ICASSP-03, Hong Kong*.

D. Klein and C. D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. Technical report.

S. Levinson. 1983. *Pragmatics*. Cambridge University Press.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proc. of ICML*.

A. Pomerantz. 1984. Agreeing and disagreeing with assessments: some features of preferred/dispreferred turn shapes. In J.M. Atkinson and J.C. Heritage, editors, *Structures of Social Action*, pages 57–101.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proc. of EMNLP*.

D. Ravichandran, E. Hovy, and F. J. Och. 2003. Statistical QA - classifier vs re-ranker: What's the difference? In *Proc. of the ACL Workshop on Multilingual Summarization and Question Answering*.

E. A. Schegloff and H Sacks. 1973. Opening up closings. *Semiotica*, 7-4:289–327.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *SIGdial Workshop on Discourse and Dialogue*, pages 97–100.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.