Argumentative Human Computer Dialogue for Automated Persuasion

Pierre Andrews* and **Suresh Manandhar*** and **Marco De Boni****

* Department of Computer Science
University of York
York YO10 5DD
UK
{pandrews,suresh}@cs.york.ac.uk

** Unilever Corporate Research
Bedford MK44 1LQ
UK
Marco.De-Boni@unilever.com

Abstract

Argumentation is an emerging topic in the field of human computer dialogue. In this paper we describe a novel approach to dialogue management that has been developed to achieve persuasion using a textual argumentation dialogue system. The paper introduces a layered management architecture that mixes task-oriented dialogue techniques with chatbot techniques to achieve better persuasiveness in the dialogue.

1 Introduction

Human computer dialogue is a wide research area in Artificial Intelligence. Computer dialogue is now used at production stage for applications such as tutorial dialogue – that helps teaching students (Freedman, 2000) – task-oriented dialogue – that achieves a particular, limited task, such as booking a trip (Allen et al., 2000) – and chatbot dialogue (Levy et al., 1997) – that is used within entertainment and help systems.

None of these approaches use persuasion as a mechanism to achieve dialogue goals. However, research towards the use of persuasion in Human Computer Interactions has spawned around the field of natural argumentation (Norman and Reed, 2003). Similarly research on Embodied Conversational Agents (ECA) (Bickmore and Picard, 2005) is also attempting to improve the persuasiveness of agents with persuasion techniques; however, it concentrates on the visual representation of the interlocutor rather than the dialogue management. Previous research on human computer dialogue has rarely focused on persuasive techniques (Guerini, Stock, and Zancanaro, 2004, initiated some research in that field). Our dialogue management system applies a novel method, taking advantage of persuasive and argumentation techniques to achieve persuasive dialogue.

According to the *cognitive dissonance* theory (Festinger, 1957), people will try to minimise the discrepancy between their behaviour and their beliefs by integrating new beliefs or distorting existing ones. In this paper, we approach persuasion as a process shaping user's beliefs to eventually change their behaviour.

The presented dialogue management system has been developed to work on known limitations of current dialogue systems:

The *impression of lack of control* is an issue when the user is interacting with a purely task-oriented dialogue system (Farzanfar et al., 2005). The system follows a plan to achieve the particular task, and the user's dialogue moves are dictated by the planner and the plan operators.

The *lack of empathy* of computers is also a problem in human-computer interaction for applications such as health-care, where persuasive dialogue could be applied (Bickmore and Giorgino, 2004). The system does not respond to the user's personal and emotional state, which sometimes lowers the user's implication in the dialogue. However, existing research (Klein, Moon, and Picard, 1999) shows that a system that gives appropriate response to the user's emotion can lower frustration.

In human-human communication, these limitations reduce the effectiveness of persuasion

Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue, pages 138–147,
Columbus, June 2008.

(Stiff and Mongeau, 2002). Even if the response towards the computer is not always identical to the one to humans, it seems sensible to think that persuasive dialogue systems can be improved by applying known findings from human-human communication.

The dialogue management architecture described in this paper (see Figure 1) addresses these dialogue management issues by using a novel layered approach to dialogue management, allowing the mixing of techniques from task-oriented dialogue management and chatbot techniques (see Section 4).

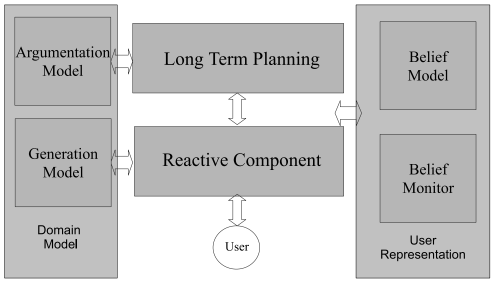

Figure 1: Layered Management Architecture

The use of a planner guarantees the consistency of the dialogue and the achievement of persuasive goals (see Section 4.2). Argumentative dialogue can be seen as a form of task-oriented dialogue where the system's task is to persuade the user by presenting the arguments. Thus, the dialogue manager first uses a task-oriented dialogue methodology to create a dialogue plan that will determine the content of the dialogue. The planning component's role is to guarantee the consistency of the dialogue and the achievement of the persuasive goals.

In state-of-the-art task-oriented dialogue management systems, the planner provides instructions for a surface realizer (Green and Lehman, 2002), responsible of generating the utterance corresponding to the plan step. Our approach is different to allow more reactivity to the user and give a feeling of control over the dialogue. In this layered approach, the reactive component provides a direct reaction to the user input, generating one or more utterances for a given plan step, allowing for reactions to user's counter arguments as well as backchannel and chitchat phases without cluttering the plan.

Experimental results show that this layered approach allows the user to feel more comfortable in the dialogue while preserving the dialogue consistency provided by the planner. Eventually, this translates into a more persuasive dialogue (see Section 6).

2 Related Work

Persuasion through dialogue is a novel field of Human Computer Interaction. Reiter, Robertson, and Osman (2003),Reed (1998) and Carenini and Moore (2000) apply persuasive communication principles to natural language generation, but only focus on monologue.

The 3-tier planner for tutoring dialogue by Zinn, Moore, and Core (2002) provides a dialogue management technique close to our approach: a top-tier generates a dialogue plan, the middle-tier generates refinements to the plan and the bottom-tier generates utterances. Mazzotta, de Rosis, and Carofiglio (2007) also propose a planning framework for user-adapted persuasion where the plan operators are mapped to natural language (or ECA) generation. However, these planning approaches do not include a mechanism to react to user's counter arguments that are difficult to plan beforehand. This paper propose a novel approach that could improve the user's comfort in the dialogue as well as its persuasiveness.

3 Case Study

Part of the problem in evaluating persuasive dialogue is using an effective evaluation framework. Moon (1998) uses the Desert Survival Scenario to evaluate the difference of persuasion and trust in interaction between humans when face-to-face or when mediated by a computer system (via an instant messaging platform).

The Desert Survival Scenario (Lafferty, Eady, and Elmers, 1974) is a negotiation scenario used in team training. The team is put in a scenario where they are stranded in the desert after a plane crash. They have to negotiate a ranking of the most eligible items (knife, compass, map, etc.) that they should keep for their survival.

For the evaluation of the dialogue system, a similar scenario is presented to the participants. The user has to choose an initial preferred ranking of items and then engages in a discussion with the dialogue system that tries to persuade the user to change the ranking. At the end of the dialogue, the user has the opportunity to either change or keep the ranking.

The architecture of the dialogue system is described throughout this paper using examples from the Desert Scenario. The full evaluation protocol is described in Section 5 and 6.

4 Dialogue Management Architecture

The following sections provide a description of the dialogue management architecture introduced in Figure 1.

4.1 Argumentation Model

The Argumentation model represents the different arguments (conclusions and premises) that can be proposed by the user or by the system. Figure 2 gives a simplified example of the Desert Scenario model.

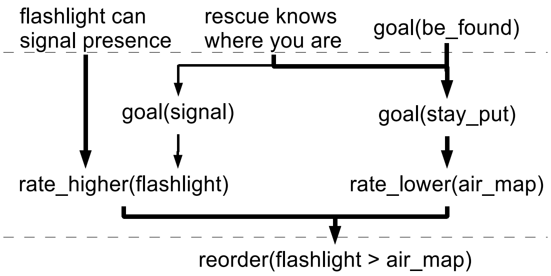

Figure 2: Argumentation Model Sample

This model shows the different facts that are known by the system and the relations between them. Arrows represent the *support* relation between two facts. For example, *rescue_knows_where_you_are* is a support to the fact ***goal****(signal)* (the user goal is to signal presence to the rescue) as well as a support to ***goal****(stay_put)* (the user goal is to stay close to the wreckage). This relational model is comparable to the argumentation framework proposed by Dung (1995), but stores more information about each argument for reasoning within the planning and reactive component (see Section 4.2).

Each fact in this model represents a belief to be introduced to the user. For example, when the dialogue tries to achieve the goal ***reorder****(flashlight > air_map)*: the system wants the user to believe that the "flashlight" item should be ranked higher than the "air_map" item. The argumentation model describes the argumentation process that is required to introduce this new belief: the system first has to make sure the user believes in ***rate_lower****(air_map)* and ***rate_higher****(flashlight)*.

Lower level facts (see Figure 2) are the goal facts of the dialogue, the ones the system chooses as dialogue goals, according to known user beliefs and the system's goal beliefs (e.g. according to the ranking the system is trying to defend). The facts in the middle of the hierarchy are intermediate facts that need to be asserted during the dialogue. The top-level facts are world knowledge: facts that require minimum defense and can be easily grounded in the dialogue.

4.2 Planning Component

The planning component's task is to find a plan using the argumentation model to introduce the required facts in the user's belief to support the persuasive goals. The plan is describes a path in the argumentation model beliefs hierarchy that translates to argumentation segments in the dialogue.

In our current evaluation method, the goal of the dialogue is to change the user's beliefs about the items so that the user eventually changes the ranking. At the beginning of the dialogue, the ranking of the system is chosen and persuasive goals are computed for the dialogue. These persuasive goals correspond to the lower level facts in the argumentation model – like "***reorder****(flashlight > air_map)*" in our previous example. The available planning operators are:

use_world*(fact)* describes a step in the dialogue that introduces a simple fact to the user.

ground*(fact)* describes a step in the dialogue that grounds a fact in the user beliefs. Grounding a fact is a different task from the ***use_world*** operator as it will need more support during the dialogue.

do_support*([fact0, fact1, ...], fact2)* describes a complex support operation. The system will initiate a dialogue segment supporting *fact2* with the facts *fact1* and *fact0*, etc. that have previously been introduced in the user beliefs.

The planning component can also use two non-argumentative operators, *do_greetings* and *do_farewells*, that are placed respectively at the beginning and the end of the dialogue plan to open and close the session.

Here is an example plan using the two arguments described in Figure 2 to support the goal ***reorder(flashlight > air map)***:

Step 1 `do_greetings`

Step 2 `use_world(goal(be_found))`
`ground(rescue_knows_where_you_are)`
`ground(can(helpatnight, item(flashlight)))`

Step 3 `do_support([can(helpatnight, item(flashlight))], rate_higher(item(flashlight)))`
`do_support( [rescue_knows_where_you_are, goal(be_found)], goal(stay_put))`

Step 4 `do_support([goal(stay_put)], rate_lower(item(air_map)))`

Step 5 `do_support(..., reorder(item(flashlight), item(air_map)))`

Step 6 `do_farewells`

The plan is then interpreted by the reactive component that is responsible for realizing each step in a dialogue segment.

4.3 The Reactive Component

The reactive component's first task is to realize the operators chosen by the planning component into dialogue utterance(s). However, it should not be mistaken for a surface language realizer. The reactive component's task, when realizing the operator, is to decide how to present the particular argumentation operator and its parameters to the user according to the dialogue context and the user's reaction to the argument. This reactive process is described in the following sections.

4.3.1 Realization and Reaction Strategies

Each step of the plan describes the general topic of a dialogue segment[1]. A dialogue segment is a set of utterances from the system and from the user that are related to a particular argument. For example, in the Desert Scenario, the operator ***ground(can(helpatnight, item(flashlight)))*** may result in the following set of utterances:

S(ystem) `I think the flashlight could be useful as it could help us at night,`

U(ser) `How is that? We are not going to move during the night.`

S `well, if we want to collect water, it will be best to do things at night and not under the burning sun.`

U `I see. It could be useful then.`

In this example, the ground operator has been realized by the reactive component in two different utterances to react to the user's interaction.

The goal of the reactive component is to make the user feel that the system understands what has been said. It is also important to avoid replanning as it tries to defend the arguments chosen in the plan.

As described in Section 4.2, the planner relies on the argumentation model to create a dialogue plan. Encoding all possible defenses and reactions to the user directly in this model will explode the search space of the planner and require careful authoring to avoid planning inconsistencies[2]. In addition, predicting at the planning level what counter arguments a user is likely to make requires a prior knowledge of the user's beliefs. At the beginning of a one-off dialogue, it is not possible to make prior assumptions on the user's beliefs; the system has a shallow knowledge of the user's beliefs and will discover them as the dialogue goes.

Hence, it is more natural to author a reactive dialogue that will respond to the user's counter arguments as they come and extends the user beliefs model as it goes. In our architecture if the user is disagreeing with an argument, the plan is not revised directly; if possible, the reactive component selects new, contextually appropriate, supporting facts for the current plan operator. It can do this multiple consecutive *local repairs* if the user needs more convincing and the domain model provides enough defenses. This allows for a simpler planning framework.

[1] i.e. it is not directly interpreted as an instruction to generate one unique utterance.

[2] a new plan could go against the previously used arguments.

In addition, when available, and even if the user agrees with the current argument, the reactive component can also choose from a set of "dialogue smoothing" or backchannel utterances to make the dialogue feel more natural. Here is an example from the Desert Scenario:

S `We don't have much water, we need to be rescued as soon as possible.` *(from plan step: **user_world**(goal(be_found)))*

U `right`

S `I am glad we agree.`*(backchannel)*

S `There is a good chance that the rescue team already knows our whereabouts. We should be optimistic and plan accordingly, don't you think?` *(from plan step: **use_world**(rescue_knows_where_you_are))*

4.3.2 Detecting user reactions

The reactive component needs to detect if the user is agreeing to its current argument or resisting the new fact that is presented. Because the dialogue management system was developed from the perspective of a system that could be easily ported to different domains, choice was made to use a domain independent and robust agreement/disagreement detection.

The agreement/disagreement detection is based on an utterance classifier. The classifier is a cascade of binary Support Vector Machines (SVM) (Vapnik, 2000) trained on the ICSI Meeting corpus (Janin et al., 2003). The corpus contains 8135 spurts[3] annotated with agreement/disagreement information Hillard, Ostendorf, and Shriberg (2003).

A multi-class SVM classifier is trained on *local features* of the spurts such as a) the length of the spurt, b) the first word of the spurt, c) the bigrams of the spurts, and d) part of speech tags. The classification achieves an accuracy of 83.17% with an N-Fold 4 ways split cross validation. Additional results and comparison with state-of-the-art are available in Appendix A.

During the dialogue, the classifier is applied on each of the user's utterances, trying to determine if the user is agreeing or disagreeing with the system. According to this labelling, the strategies described in section 4.3.1 and 4.3.3 are applied.

4.3.3 Revising the plan

The reactive component will attempt *local repairs* to the plan by defending the argumentation move chosen by the planning component. However, there are cases when the user will still not accept an argument. In these cases, imposing the belief to the user is counter-productive and the current goal belief should be dropped from the plan.

For each utterance chosen by the reactive component, the belief model of the user is updated to represent the system knowledge of the user's beliefs. Every time the user agrees to an utterance from the system, the belief model is extended with a new belief; in the previous example, when the user says *"I see, it could be useful then."*, the system detects an agreement (see the Section 4.3.2) and extends the user's beliefs model with the belief: ***can**(helpatnight, item(flashlight))*. The agreement is then followed by a *local repair*, since the user doesn't disagree with the statement made, the system also extends the belief model with beliefs relevant to the content of the local repair, thus learning more about the user's belief model.

As a result of this process, when the system decides to revise the plan, the planning component does not start from the same beliefs state as previously. In effect, the system is able to learn user's beliefs based on the agreement/disagreement with the user, it can therefore make a more effective use of the argumentation hierarchy to find a better plan to achieve the persuasive goals.

Still, there are some cases when the planning component will be unable to find a new plan from the current belief state to the goal belief state – this can happen when the planner has exhausted all its argumentative moves for a particular sub-goal. In these cases, the system has to make concessions and drop the persuasive goals that it cannot fulfil. By dropping goals, the system will lower the final persuasiveness, but guarantees not coercing the user.

4.3.4 Generation

Utterance generation is made at the reactive component level. In the current version of the dialogue management system, the utterance generation

[3]speech utterances that have no pauses longer than .5 seconds.

is based on an extended version of Alicebot AIML [4].

AIML is an XML language that provides a pattern/template generation model mainly used for chatbot systems. An AIML bot defines a set of categories that associate a *topic*, the context of the previous bot utterance (called *that* in the AIML terminology), a *matching pattern* that will match the last user utterance and a *generation template*. The *topic*, *matching* and *that* field define matching patterns that can contain * wildcards accepting any token(s) of the user utterance (e.g. *HELLO* * would match any utterance starting by "Hello"). They are linked to a *generation template* that can reuse the tokens matched by the patterns wildcards to generate an utterance tailored to the user input and the dialogue context.

For the purpose of layered dialogue management, the AIML language has been extended to include more features: 1) A new pattern slot has been introduced to link a set of categories to a particular argumentation operator; 2) Utterances generations are linked to the belief they are trying to introduce to the user and if an agreement is detected, this belief is added to the user belief model.

For example, a set of matching categories for the Desert Scenario could be:

Plan operator: `use_world(goal(survive))`

- **Category 1** :
  - **Pattern** `*`
  - **Template** `Surviving is our priority, do you want to hear about my desert survival insights?`
- **Category 2** :
  - **Pattern** `* insights`
  - **That** `* survival insights`
  - **Template** `I mean, I had a few ideas ...common knowledge I suppose.`
- **Category 3** :
  - **Pattern** `*`
  - **That** `* survival insights`
  - **Template** `Well, we are in this together. Let me tell you of what I think of desert survival, ok?`

[4] `http://www.alicebot.org/`

These three categories can be used to match the user reaction during the dialogue segment corresponding to the plan operator: ***use_world****(goal(survive)).* *Category 1* is used as the initiative taking generation. It will be the first one to be used when the system comes from a previously finished step. *Categories 2-3* are all "defenses" that support *Category 1.* They will be used to react to the user if no agreement is detected from the last utterances. For example, if the user says *"what kind of survival insights??"* as a reply to the generation from *Category 1*, a disagreement is detected and the reactive component will have a contextualised answer as given by *category 2* whose `that` pattern matches the last utterance from the system, the `pattern` pattern matches the user utterance.

The dialogue management system uses 187 categories tailored to the Desert Scenario as well as 3737 general categories coming from the Alice chatbot and used to generate the dialogue smoothing utterances. Developing domain specific reactions is a tedious and slow process that was iteratively achieved with Wizard of OZ experiments with real users. In these experiments, users were told they were going to have a dialogue with another human in the Desert Scenario context. The dialogue system manages the whole dialogue, except for the generation phase that is mediated by an expert that can either choose the reaction of the system from an existing set of utterances, or type a new one.

5 Persuasiveness Metric

Evaluating a behavior change would require a long-term observation of the behavior that would be dependent to external elements (Bickmore and Picard, 2005). To evaluate our system, an evaluation protocol measuring the change in the beliefs underlying the behavior was chosen. As explained in Section 3, the Desert Scenario is used as a base for the evaluation. Each participant is told that he is stranded in the desert. The user gives a preferred initial ranking R_i of the items (knife, compass, map, etc.). The user then engages in a dialogue with the system. The system then attempts to change the user's ranking to a different ranking R_s through persuasive dialogue. At the end of the dialogue, the user can change this choice to arrive at a final ranking R_f.

The persuasiveness of the dialogue can be measured as the evolution of the distance between the user ranking (R_i, R_f) and the system ranking (R_s). The Kendall τ distance (Kendall, 1938) is used to compute the pairwise disagreement between two rankings. The change of the Kendall τ distance during the dialogue gives an evaluation of the persuasiveness of the dialogue: $P_{ersuasiveness} = K\tau(R_i, R_s) - K\tau(R_f, R_s)$. In the current evaluation protocol, the R_s is always the reverse of the R_i, so $K\tau(R_i, R_s)$ is always the maximum distance possible: $\frac{n\times(n-1)}{2}$ where n is the number of items to rank. The minimum Kendall tau distance is 0. If the system was persuasive enough to make the user invert the initial ranking, $P_{ersuasiveness}$ of the system is maximum and equal to: $\frac{n\times(n-1)}{2}$. If the system does not succeed in changing the user ranking, then $P_{ersuasiveness}$ is zero.

6 Evaluation Results and Discussion

16 participants have been recruited from a variety of ages (from 20 to 59) and background. They were all told to use a web application that describes the Desert Scenario (see Section 3) and proposes to undertake two instant messaging chats with two human users[5]. However, both discussions are managed by different versions of the dialogue system, following a similar protocol:

- one version of the dialogue is managed by a *limited* version of the dialogue system, with no reactive component. This version is similar to a purely task-oriented system, planning and revising the plan directly on dialogue failures,
- the second version is the *full* dialogue system as described in this paper.

Each participant went through one dialogue with each system, in a random order. This comparison shows that the dialogue flexibility provided by the reactive component allows a more persuasive dialogue. In addition, when faced with the second dialogue, the participant has formed more beliefs about the scenario and is more able to counter argue.

[5]The evaluation is available Online at `http://www.cs.york.ac.uk/aig/eden`

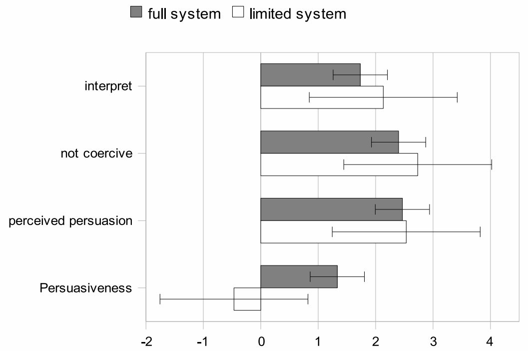

Figure 3: Comparative Results. *interpret, not coercive, perceived persuasion* are on a scale of $[0-4]$ (see Appendix B). $P_{ersuasiveness}$ is on a scale of $[-10, 10]$.

Figure 3 reports the independent $P_{ersuasiveness}$ metric results as well as interesting answers to a questionnaire that the participants filled after each dialogue (see the Appendix B for detailed results and questionnaire).

Over all the dialogues, the *full* system is **18%** more persuasive than the *limited* system. This is measured by the $P_{ersuasiveness}$ metric introduced in Section 5. With the *full* system, the participants did an average of **1.33 swaps of items towards** the system's ranking. With the *limited* system, the participants did an average of **0.47 swaps of items away** from the system's ranking. However, the answers to the self evaluated *perceived persuasion* question show that the participants did not see any significant difference in the ability to persuade of the *limited* and the *full* systems.

According to the question *interpret*, the participants found that the *limited* system understood better what they said. This last result might be explained by the behavior of the systems: the *limited* system drops an argument at every user disagreement, making the user believe that the disagreement was understood. The *full* system tries to defend the argument; if possible with a contextually tailored support, however, if this is not available, it may use a generic support, making the user believe he was not fully understood.

Our interpretation of the fact that the discrepancy between user self evaluation of the interaction with the system and the measured persuasion is that, even if the *full* system is more argumentative, the user didn't feel coerced[6]. These results show that a more persuasive dialogue can be achieved without deteriorating the user perception of the interaction.

7 Conclusion

Our dialogue management system introduces a novel approach to dialogue management by using a layered model mixing the advantages of state-of-the-art dialogue management approaches. A planning component tailored to the task of argumentation and persuasion searches the ideal path in an argumentation model to persuade the user. To give a reactive and natural feel to the dialogue, this task-oriented layer is extended by a reactive component inspired from the chatbot dialogue management approach. The Desert Scenario evaluation, providing a simple and independent metric for the persuasiveness of the dialogue system provided a good protocol for the evaluation of the dialogue system. This one showed to be 18% more persuasive than a purely task-oriented system that was not able to react to the user interaction as smoothly.

Our current research on the dialogue management system consists in developing another evaluation domain where a more complex utterance generation can be used. This will allow going further than the simple template based system, offering more diverse answers to the user and avoiding repetitions; it will also allow us to experiment textual persuasion tailored to other parameters of the user representation, such as the user personality.

References

Allen, J. F., G. Ferguson, B. W. Miller, E. K. Ringger, and T. Sikorski. 2000. *Dialogue Systems: From Theory to Practice in TRAINS-96*, chapter 14.

Bickmore, T. and T. Giorgino. 2004. Some novel aspects of health communication from a dialogue systems perspective. In *AAAI Fall Symposium*.

Bickmore, T. W. and R. W. Picard. 2005. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Comput.-Hum. Interact.*, 12(2):293–327.

Carenini, G. and J. Moore. 2000. A strategy for generating evaluative arguments. In *International Conference on Natural Language Generation*.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357.

Farzanfar, R., S. Frishkopf, J. Migneault, and R. Friedman. 2005. Telephone-linked care for physical activity: a qualitative evaluation of the use patterns of an information technology program for patients. *J. of Biomedical Informatics*, 38(3):220–228.

Festinger, Leon. 1957. *A Theory of Cognitive Dissonance*. Stanford University Press.

Freedman, R. 2000. Plan-based dialogue management in a physics tutor. In *Proceedings of ANLP '00*.

Galley, M., K. Mckeown, J. Hirschberg, and E. Shriberg. 2004. Identifying agreement and disagreement in conversational speech: use of bayesian networks to model pragmatic dependencies. In *Proceedings of ACL'04*.

Green, N. and J. F. Lehman. 2002. An integrated discourse recipe-based model for task-oriented dialogue. *Discourse Processes*, 33(2):133–158.

Guerini, M., O. Stock, and M. Zancanaro. 2004. Persuasive strategies and rhetorical relation selection. In *Proceedings of ECAI-CMNA*.

Hillard, D., M. Ostendorf, and E. Shriberg. 2003. Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proceedings of NAACL'03*.

Janin, A., 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 *Proceedings of ICASSP'03*.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Klein, J., Y. Moon, and R. W. Picard. 1999. This computer responds to user frustration. In *CHI'99*.

Lafferty, J. C., Eady, and J. Elmers. 1974. *The desert survival problem*.

Levy, D., R. Catizone, B. Battacharia, A. Krotov, and Y. Wilks. 1997. Converse:a conversational companion. In *Proceedings of 1st International Workshop on Human-Computer Conversation*.

Mazzotta, I., F. de Rosis, and V. Carofiglio. 2007. Portia: A user-adapted persuasion system in the healthy-eating domain. *Intelligent Systems, IEEE*, 22(6).

Moon, Y. 1998. The effects of distance in local versus remote human-computer interaction. In *Proceedings of SIGCHI'98*.

Norman, Timothy J. and Chris Reed. 2003. *Argumentation Machines : New Frontiers in Argument and Computation (Argumentation Library)*. Springer.

Reed, C. 1998. *Generating Arguments in Natural Language*. Ph.D. thesis, University College London.

Reiter, E., R. Robertson, and L. M. Osman. 2003. Lessons from a failure: generating tailored smoking cessation letters. *Artif. Intell.*, 144(1-2):41–58.

Stiff, J. B. and P. A. Mongeau. 2002. *Persuasive Communication*, second edition.

Vapnik, V. N. 2000. *The Nature of Statistical Learning Theory*.

Zinn, C., J. D. Moore, and M. G. Core. 2002. A 3-tier planning architecture for managing tutorial dialogue. In *Proceedings of ITS '02*.

[6]The answers to the *not coercive* question do not show any significant difference in the perception of coercion of the two system.

A Agreement/Disagreement Classification

	Setup 1	Setup 2
Galley et al., global features	86.92%	84.07%
Galley et al., local features	85.62%	83.11%
Hillard et al.	82%	NA
SVM	**86.47%**	**83.17%**

Table 1: Accuracy of different agreement/disagreement classification approaches.

The accuracy of state-of-the-art techniques (Hillard, Ostendorf, and Shriberg (2003) and Galley et al. (2004)) are reported in Table 1 and compared to our SVM classifier. Two experimental setups were used:

Setup 1 reproduces Hillard, Ostendorf, and Shriberg (2003) training/testing split between meetings;

Setup 2 reproduces the N-Fold, 4 ways split used by Galley et al. (2004).

The SVM results are arguably lower than Galley et al. system with labeled dependencies. However, this is because our system only relies on local features of each utterance, while Galley et al. (2004) use *global features* (i.e. features describing relations between consecutive utterances) suggest that adding global features would also improve the SVM classifier.

B Evaluation Questionnaire

In the evaluation described in section 6, the participants were asked to give their level of agreement with each statement on the scale: Strongly disagree (0), Disagree (1), Neither agree nor disagree (2), Agree (3), Strongly Agree(4). Table 2 provides a list of questions with the average agreement level and the result of a paired t-test between the two system results.

label	question	*full* system	*limited* system	ttest
interpret	"In the conversation, the other user interpreted correctly what you said"	1.73	2.13	0.06
perceived persuasion	"In the conversation, the other user was persuasive"	2.47	2.53	0.44
not coercive	"The other user was not forceful in changing your opinion"	2.4	2.73	0.15
sluggish	"The other user was sluggish and slow to reply to you in this conversation"	1.27	1.27	0.5
understand	"The other user was easy to understand in the conversation"	3.2	3.13	0.4
pace	"The pace of interaction with the other user was appropriate in this conversation"	2.73	3.07	0.1
friendliness	"The other user was friendly"	2.93	2.87	0.4
length	length of the dialogue	12min 19s	08min 33s	0.07
persuasiveness	$P_{ersuasiveness}$	1.33	-0.47	0.05

Table 2: Results from the evaluation questionnaire.