An Information State-Based Dialogue Manager for Call for Fire Dialogues

Antonio Roque and **David Traum**
USC Institute for Creative Technologies
13274 Fiji Way, Marina Del Rey, CA 90292
roque@ict.usc.edu, traum@ict.usc.edu

Abstract

We present a dialogue manager for "Call for Fire" training dialogues. We describe the training environment, the domain, the features of its novel information state-based dialogue manager, the system it is a part of, and preliminary evaluation results.

1 Overview

Dialogue systems are built for many different purposes, including information gathering (e.g., (Aust et al., 1995)), performing simple transactions (e.g, (Walker and Hirschman, 2000)), collaborative interaction (e.g., (Allen et al., 1996)), tutoring (e.g., (Rose et al., 2003)), and training (e.g. (Traum and Rickel, 2002)). Aspects of the purpose, as well as features of the domain itself (e.g., train timetables, air flight bookings, schedule maintenance, physics, and platoon-level military operations) will have a profound effect on the nature of the dialogue which a system will need to engage in. Issues such as initiative, error correction, flexibility in phrasing and dialogue structure may depend crucially on these factors.

The information state approach to dialogue managers (Larsson and Traum, 2000) has been an attempt to cast some of these differences within the same framework. In this approach, a theory of dialogue is constructed by providing information structure elements, a set of dialogue moves that can be recognized and produced and are used to modify the nature of these elements, a set of update rules that govern the dynamics of how the information is changed as dialogue moves are performed, and an update strategy. Many different dialogue systems have been built according to this general approach (e.g., (Cooper and Larsson, 1999; Matheson et al., 2000; Lemon et al., 2001; Johnston et al., 2002; Traum and Rickel, 2002; Purver, 2002)).

In this paper, we present an information-state based dialogue manager for a new domain: training call for fire dialogues. Like other dialogue systems used as role-players in training applications, the structure of the dialogue is not completely free for a dialogue designer to specify based on issues of dialogue efficiency. The dialogue system must conform as much as possible to the type of dialogue that a trainee would actually encounter in the types of interaction he or she is being trained for. In particular, for military radio dialogues, much of the protocol for interaction is specified by convention (e.g., (Army, 2001)). Still, there is a fair amount of flexibility in how other aspects of the dialogue progress.

This dialogue manager is part of a system we call Radiobot-CFF. Radiobots are a general class of dialogue systems meant to speak over the radio in military simulations. Our most extended effort to date is the Radiobot-CFF system, which engages in "call for fire" dialogues to train artillery observers within a virtual reality training simulation. Our dialogue system can operate according to three different use cases, depending on how much control a human operator/trainer would like to exercise over the dialogue. There is a fully automatic mode in which the Radiobot-CFF system engages unassisted in dialogue with the user, a semi-automatic mode in which the Radiobot-CFF system fills in forms (which can be edited) and the operator can approve or change communication with a simulator or trainee, and a passive mode in which the operator is engaging in the dialogue and the Radiobot-CFF system is just observing.

In section 2, we describe the training applica-

Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, pages 88–95,
Sydney, July 2006. ©2006 Association for Computational Linguistics

tion that our dialogue system has been embedded in as well as the system itself. In section 3, we describe some aspects of "call for fire dialogues", especially the differences in initiative and purposes of different phases in the dialogue. In section 4, we describe the information-state based dialogue model we have developed for this domain. This includes dialogue moves, information components, and update rules. We describe some error handling capabilities in section 5, and evaluation results in section 6.

2 Testbed

Our current testbed, Radiobot-CFF, has been developed in a military training environment, JFETS-UTM, at the U.S. Army base in in Ft. Sill, Oklahoma. JFETS-UTM trains soldiers to make Calls for Fire (CFFs), in which a Forward Observer (FO) team locates an enemy target and requests an artillery fire mission by radio from a Fire Direction Center (FDC). The training room resembles a battle-scarred apartment in a Middle Eastern country. A window shows a virtual city displayed by a rear-projected computer screen, and the soldiers use binoculars with computer displays at their ends to search for targets.

Ordinarily, two trainers control a UTM session. One communicates with the FO via a simulated radio, and the other decides what the artillery fire should be and inputs it to a GUI for the simulator. It is our goal to replace those two trainers with one trainer focusing on assessment while Radiobot-CFF handles the radio communications and interfaces with the virtual world.

Radiobot-CFF is composed of several pipelined components. A Speech Recognition component is implemented using the SONIC speech recognition system (Pellom, 2001) with custom language and acoustic models. An Interpreter component tags the ASR output with its its dialogue move and parameter labels using two separate Conditional Random Field (Sha and Pereira, 2003; McCallum, 2002) taggers trained on hand-annotated utterances. A Dialogue Manager processes the tagged output, sending a reply to the FO (via a template-based Generator) and, when necessary, a message to the artillery simulator FireSim XXI[1] to make decisions on what type of fire to send. The reply to FO and messages to simulator are mediated by GUIs where the trainer can intervene if need be.

3 Call for Fire Dialogues

Call for Fire procedures are specified in an Army field manual (Army, 2001) with variations based on a unit's standard operating procedure. Messages are brief and followed by confirmations, where any misunderstandings are immediately corrected. A typical CFF is shown in Figure 1.

1	FO	steel one niner this is gator niner one adjust fire polar over
2	FDC	gator nine one this is steel one nine adjust fire polar out
3	FO	direction five niner four zero distance four eight zero over
4	FDC	direction five nine four zero distance four eight zero out
5	FO	one b m p in the open i c m in effect over
6	FDC	one b m p in the open i c m in effect out
7	FDC	message to observer kilo alpha high explosive four rounds adjust fire target number alpha bravo one zero zero zero over
8	FO	m t o kilo alpha four rounds target number alpha bravo one out
9	FDC	shot over
10	FO	shot out
11	FDC	splash over
12	FO	splash out
13	FO	right five zero fire for effect out over
14	FDC	right five zero fire for effect out
15	FDC	shot over
16	FO	shot out
17	FDC	rounds complete over
18	FO	rounds complete out
19	FO	end of mission one b m p suppressed zero casualties over
20	FDC	end of mission one b m p suppressed zero casualties out

Figure 1: Example Dialogue with Radiobot-CFF

CFFs can generally be divided into three *phases*. In the first phase (utterances 1-6 in Figure 1) the FOs identify themselves and important information about the CFF, including their coordinates, the kind of fire they are requesting, the location of the target, and the kind of target. In utterance 1 in Figure 1 the FO performs an identification, giving his own call sign and that of the FDC he is calling, and also specifies a method of fire ("adjust fire") and a method of targeting ("polar".) Note that when speakers expect a reply, they end their utterance with "over" as in utterance 1, otherwise with "out" as in the confirmation in utterance 2. In utterance 3 the FO gives target coordinates, and in utterance 5 the FO identifies the target as a BMP (a type of light tank) and requests ICM rounds ("improved conventional munitions".) These turns typically follow one another

[1] http://sill-www.army.mil/blab/sims/FireSimXXI.htm

in quick sequence.

In the second phase of a CFF, (utterances 7-12 in Figure 1), after the FDC decides what kind of fire they will send, they inform the FO in a message to observer (MTO) as in utterance 7. This includes the units that will fire ("kilo alpha"), the kind of ammunition ("high explosive"), the number of rounds and method of fire ("4 rounds adjust fire"), and the target number ("alpha bravo one zero zero zero"). CFFs are requests rather than orders, and they may be denied in full or in part. In this example, the FO's request for ICM rounds was denied in favor of High Explosive rounds. Next the FDC informs the FO when the fire mission has been shot, as in utterance 9, and when the fire is about to land, as in utterance 11. Each of these are confirmed by the FO.

In the third phase, (utterances 13-20 in Figure 1) the FO regains dialogue initiative. Depending on the observed results, the FO may request that the fire be repeated with an adjust in location or method of fire. In utterance 13 the FO requests that the shot be re-sent to a location 50 meters to the right of the previous shot as a "fire for effect" all-out bombardment rather than an "adjust fire" targeting fire. This is followed by the abbreviated FDC-initiated phase of utterances 15-18. In utterance 19 the FO ends the mission, describing the results and number of casualties.

Besides the behavior shown, at any turn either participant may request or initiate an intelligence report or request the status of a mission. Furthermore, after receiving an MTO the FO may immediately begin another fire mission and thus have multiple missions active; subsequent adjusts are disambiguated with the target numbers assigned during the MTOs.

4 Dialogue Manager

We have constructed an Information State-based dialogue manager (Larsson and Traum, 2000) on this domain consisting of a set of dialogue moves, a set of informational components with appropriate formal representations, and a set of update rules with an update strategy. We describe each of these in turn.

4.1 Dialogue Moves

We defined a set of dialogue moves to represent the incoming FO utterances based on a study of transcripts of human-controlled JFETS-UTM sessions, Army manuals, and the needs of the simulator. As shown in Figure 2 these are divided into three groups: those that provide information about the FO or the fire mission, those that confirm information that the FDC has transmitted, and those that make requests.

```
Mission Information:
  Observer Coordinates
  Situation Report
  Identification
  Warning Order
  Method of Control
  Method of Engagement
  Target Location
  Target Description
  End of Mission

Confirming Information:
  Message to Observer
  Shot
  Splash
  Rounds Complete
  Intel Report

Other Requests:
  Radio Check
  Say Again
  Status
  Standby
  Command
```

Figure 2: FO Dialogue Moves

The dialogue moves that provide information include those in which the FOs transmit their Observer Coordinates (grid location on a map), a generic Situation Report, or one of the various components of a fire mission request ranging from call sign Identification to final End of Mission. The dialogue moves that confirm information include those that confirm the MTO and other FDC-initiated utterances, or a general report on scenario Intel. The final group includes requests to check radio functionality, to repeat the previous utterance, for status of a shot, to stand by for transmission of information, and finally a set of commands such as "check fire" requesting cancellation of a submitted fire mission.

Each of these dialogue moves contains information important to the dialogue manager. This information is captured by the parameters of the dialogue move, which are enumerated in Figure 3. Each parameter is listed with the dialogue move it usually occurs with, but this assignment is not strict. For example, "number_of_enemies" parameters occur in Target Description as well as End of Mission dialogue moves.

```
Identification-related:
  fdc_id
  fo_id

Warning Order-related:
  method_of_fire
  method_of_control
  method_of_engagement
  method_of_location

Target Location-related:
  grid_location
  direction
  distance
  attitude
  left_right
  left_right_adjust
  add_drop
  add_drop_adjust
  known_point

End Of Mission-related:
  target_type
  target_description
  number_of_enemies
  disposition

Other:
  command
  detail_of_request
  target_number
```

Figure 3: Dialogue Move Parameters

Figure 4 shows how the dialogue moves and parameters act to identify the components of an FO utterance. The example is based on utterance 1 in Figure 1; the Identification move has two parameters representing the call signs of the FDC and the FO, and the Warning Order has two parameters representing the method of fire and method of location. Parameters need to be identified to confirm back to the FO, and in some cases to be sent to the simulator and for use in updating the information state. In the example in Figure 4, the fact that the requested method of fire is an "adjust fire" will be sent to the simulator, and the fact that a method of fire has been given will be updated in the information state.

Identification: steel one nine this is gator niner one
`fdc_id`: steel one nine
`fo_id`: gator niner one
Warning Order: adjust fire polar
`method_of_fire`: adjust fire
`method_of_location`: polar

Figure 4: Example Dialogue Moves and Parameters

4.2 Informational Components

The Radiobot-CFF dialogue manager's information state consists of five classes of informational components, defined by their role in the dialogue and their level of accessibility to the user. These are the Fire Mission Decision components, the Fire Mission Value components, the Post-Fire Value components, the Disambiguation components, and the Update Rule Processing components.

By dividing the components into multiple classes we separate those that are simulator-specific from more general aspects of the domain.

Decisions to fire are based on general constraints of the domain, whereas the exact components to include in a message to simulator will be simulator-specific. Also, the components have been designed such that there is almost no overlap in the update rules that modify them (see section 4.3). This reduces the complexity involved in editing or adding rules; although there are over 100 rules in the information state, there are few unanticipated side-effects when rules are altered.

The first class of components are the Fire Mission Decision components, which are used to determine whether enough information has been collected to send fire. These components are boolean flags, updated by rules based on incoming dialogue moves and parameters. Figure 5 shows the values of these components after utterance 3 in Figure 1 has been processed. The FO has given a warning order, and a target location (which can either be given through a grid location, or through a combination of direction and distance values, and observer coordinates), so the appropriate components are "true". After the FO gives a target description, that component will be true as well, and an update rule will recognize that enough information has been gathered to send a fire mission.

```
has warning order? true
has target location? true
    has grid location? false
    has polar direction? true
    has polar distance? true
    has polar obco? true
has target descr? false
```

Figure 5: Fire Mission Decision Components

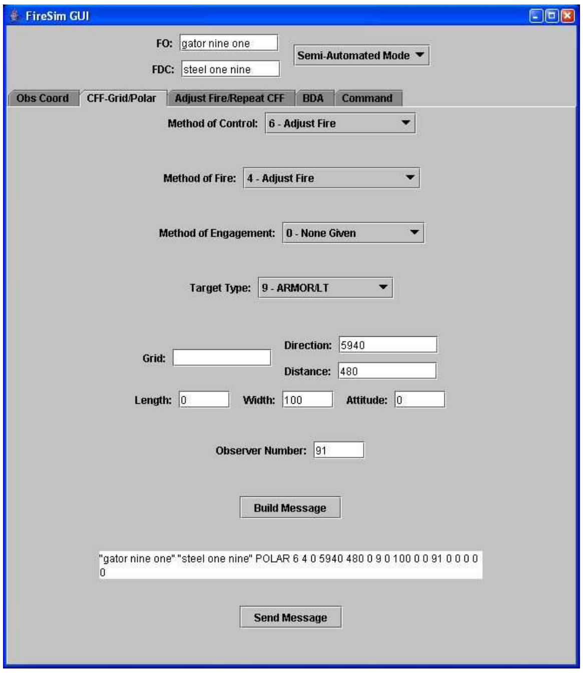
The second class of information state components is the set of Fire Mission Value components, which track the value of various information elements necessary for requesting a fire mission. These are specific to the FireSim XXI simulator. Figure 6 shows the values after utterance 3 in Figure 1. Components such as “direction value” take number values, and components such as “method of fire” take values from a finite set of possibilities. Several of these components, such as “attitude” have defaults that are rarely changed. Once the dialogue manager or human trainer decides that it has enough information to request fire, these components are translated into a simulator command and sent to the simulator.

```
method of control: adjust fire
method of fire: adjust fire
method of engagement: none given
target type: -
grid value: -
direction value: 5940
distance value: 480
length: 0
width: 100
attitude: 0
observer coordinate value: 45603595
```

Figure 6: Fire Mission Value Components

Fire Mission Value components are also directly modifiable by the trainer. Figure 7 shows the GUI which the trainer can use to take control of the session, edit any of the Fire Mission Value components, and relinquish control of the session back to Radiobot-CFF. This allows the trainer to correct any mistakes that the Radiobot may have made or test the trainee’s adaptability by sending the fire to an unexpected location. The example shown in Figure 7 is after utterance 5 of Figure 1; the system is running in semi-automated mode and the dialogue manager has decided that it has enough information to send a fire. The trainer may send the message or edit it and then send it. A second GUI, not shown, allows the trainer to take control of the outgoing speech of the Radiobot, and, in semi-automated mode, either confirm the sending of a suggested output utterance, alter it before sending, or author new text for the radiobot to say.

The third class of components is the Post-Fire Value components, which are also exposed to the trainer for modification. The example shown in Figure 8 is from after utterance 13 in Figure 1; the FO has requested an “adjust fire” with an indicator of “fire for effect” and a right adjustment of 50. At this point in the dialogue the FO could have instead chosen to end the mission. If the initial fire had been a “fire for effect” it could have been repeated, rather than following up an initial “adjust fire.” The adjust fire stage does not have any decision components because typically the adjust information is given in one move.



FO: gator nine one
FDC: steel one nine
Semi-Automated Mode

Method of Control: 6 - Adjust Fire
Method of Fire: 4 - Adjust Fire
Method of Engagement: 0 - None Given
Target Type: 9 - ARMOR/LT
Grid:
Direction: 5940
Distance: 480
Length: 0 Width: 100 Attitude: 0
Observer Number: 91

"gator nine one" "steel one nine" POLAR 6 4 0 5940 480 0 9 0 100 0 0 91 0 0 0 0
0

Figure 7: GUI

```
adjust fire: true
  shift indicator: fire for effect
repeat FFE: false

  left-right adjustment: 50
  add-drop adjustment: 0
  vertical adjustment: 0

end of mission: false
  disposition: -
  number of casualties: -
```

Figure 8: Post-Fire Value Components

The fourth class, Disambiguation components, are used by many rules to disambiguate local information based on global dialogue features. The example shown in Figure 9 is from the dialogue in Figure 1, after utterance 1. The “mission is polar” component helps determine the method of target location if speech recognition erroneously detects both polar and grid coordinates. Target numbers allow the FOs to handle multiple missions at the same time (e.g., starting a new call for fire, before the previous mission has been completed). The “missions active” component tracks how many missions are currently being discussed. The “phase” refers to the state of a three-state FSA that tracks which of the three subdialogue phases (described in section 3) the dialogue is in for the most recently-discussed mission.

An example of the use of the Disambiguation components is to determine whether the phrase “fire for effect” refers to an adjustment of a previous mission or the initiation of a new mission. In utterance 13 in Figure 1, “fire for effect” refers to an adjustment of a CFF that began with an “adjust fire” in utterance 1. However, the FO could have started that CFF by calling for a “fire for effect”. Furthermore the FO could have started a second CFF in utterance 13 rather than doing an adjust, and might have specified “fire for effect”. By using a rule to check the phase of the mission the move can be disambiguated to understand that it is referring to an adjustment, rather than the initiation of a new fire mission.

```
mission is polar?: true
target number: 0
missions active: 0
last method of fire: adjust
phase: Info-Gathering
```

Figure 9: Disambiguation Components

The last class of components, shown in Figure 10, is closely tied to the update rule processing, and is therefore described in the following section.

```
current reply: gator nine one this is
               steel one nine
previous reply: -
understood? true
send EOM? false
send repeat? false
send repeat adjust? false
send repeat ffe? false
```

Figure 10: Update Rule Processing Components

4.3 Update Rules

Update rules update the informational components, build a message to send to the FO, build a message to send to the simulator, and decide whether a message should actually be sent to the FO or simulator.

As an example of rule application, consider the processing of utterance 1 in Figure 1. Figure 4 shows the moves and parameters for this utterance. When the dialogue manager processes this utterance, a set of rules associated with the Identification move are applied, which starts building a response to the FO. This response is built in the “current reply” Update Rule Processing component. Figure 10 shows a reply in the process of being built: a rule has recognized that an Identification move is being given, and has filled in slots in a template with the necessary information and added it to the “current reply” component.

Next, the update rules will recognize that a Warning Order is being given, and will identify that it is an “adjust fire” method of fire, and update the “has warning order” decision component, the “method of control” and “method of fire” value components, and the “last method of fire” disambiguation component. As part of this, the appropriate fields of the GUIs will be filled in to allow the trainer to override the FO’s request if need be. Another rule will then fill in the slots of a template to add “adjust fire polar” to the current reply, and later another rule will add “out”, thus finishing the reply to the FO. After the reply is finished, it will place it in the “previous reply” component, for reference if the FO requests a repeat of the previous utterance.

Certain rules are specified as achieving comprehension — that is, if they are applied, the “understood” variable for that turn is set. If no reply has been built but the move has been understood, then no reply needs to be sent. This happens, for example, for each of utterances 8, 10, and 12 in Figure 1: because they are confirmations of utterances that the FDC has initiated, they do not need to be replied to. Similarly, no reply needs to be sent if no reply has been built and the incoming message is empty or only contains one or two words indicative of an open mic and background noise. Finally, if no reply has been built and the move has not been understood, then the FO is prompted to repeat the message.

As described above, the Fire Mission Decision components are used to determine whether to send a fire mission. For other communications with the simulator, a simpler approach is possible. The decisions to send an end of mission, a repeat fire, or a repeat fire with the ’adjust’ or ’fire for effect’ specification can be made with update rules acting on a single boolean, and so these are also part of the Update Rule Processing Components as shown in Figure 10.

Finally, the application of rules follows a specific strategy. A given utterance may contain one or more dialogue moves, each with a set of rules specific to it. The dialogue manager applies the appropriate rules to each dialogue move in the utterance before applying the rules that send the FO messages or simulator commands, as shown in Figure 11. Rules for producing replies and simulator commands are delayed until the end of processing an utterance to allow for utterances that may contain self-corrections or relevant details later in the turn.

```
for each dialogue move in utterance
  apply rules for that dialogue move
end for

apply rules to send reply to FO
apply rules to send simulator commands
```

Figure 11: Update Strategy for Rules

5 Error Handling

Radiobot-CFF is able to handle various kind of problematic input in a number of ways. It can handle partially correct information, as in Figure 12. Speech recognition errors caused the "three casualties" information to be lost, but the update rules were able to handle the essential part of the FO contribution: that the mission was ended, and that the target was neutralized. The domain is forgiving in this particular example, although a strict trainer might want to intervene by the GUI and insist that the FO re-submit the end of mission report.

```
FO Said:    end of mission target
            neutralized estimate three
            casualties over
ASR Output: in end of mission target
            neutralized as the make three
            catch a these over
Radiobot:   end of mission target
            neutralized out
```

Figure 12: Error Correction

In other cases, such as when giving number coordinates, all information must be fully grounded. An example of this is in Figure 13, where the number "five" is lost by the speech recognition. In this case, the domain-appropriate response is to prompt for a repetition.

```
FO Said:    right five zero over
ASR Output: right by zero over
Radiobot:   say again over
```

Figure 13: Error Correction - Prompt

6 Evaluation

We conducted an evaluation of the Radiobot-CFF system in fully-automated, semi-automated, and human-controlled conditions. The system performed well in a number of measures; for example, Table 1 shows the scores for median time-to-fire and task-completion rates. Additional measures and further details are available in (Robinson et al., 2006).

Table 1: *Example Evaluation Measures*

Measure	Human	Semi	Fully
Time To Fire	106.2 s	139.4 s	104.3 s
Task Compl.	100%	97.5%	85.9%

Of particular relevance here, we performed an evaluation of the dialogue manager, using the evaluation corpus of 17 missions run on 8 sessions, a total of 408 FO utterances. We took transcribed recordings of the FO utterances, ran them through the Interpreter, and corrected them. For each session, we ran corrected Interpreter output through the Dialogue Manager to print out the values of the informational components at the end of every turn. We then corrected those, and compared the corrections to the uncorrected values to receive precision, accuracy, and f-scores of 0.99 each.[2]

7 Summary

We presented a dialogue manager which can engage in Call for Fire training dialogues, and described the environment and system in which it works. It has an information state-based design with several components accessible to a human operator, and may be controlled either fully, in part, or not at all by that human operator.

8 Acknowledgements

This work has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

[2] In this preliminary evaluation, the Interpreter and informational component corrections were all done by a single coder; also, the coder was correcting the informational component output rather than entering informational component information from blank, thus any errors of omission on the part of the coder would work in favor of the system performance.

We would like to thank Charles Hernandez and Janet Sutton of the Army Research Laboratory, and Bill Millspaugh and the Depth & Simultaneous Attack Battle Lab in Fort Sill, Oklahoma, for their efforts on this project. We would also like to thank the other members of the Radiobots project.

References

James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski. 1996. A robust system for natural spoken dialogue. In *Proceedings of the 1996 Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 62–70.

Department of the Army. 2001. Tactics, techniques and procedures for observed fire and fire support at battalion task force and below. Technical Report FM 3-09.30 (6-30), Department of the Army.

H. Aust, M. Oerder, F. Siede, and V. Steinbiss. 1995. A spoken language enquiry system for automatic train timetable information. *Philips Journal of Research*, 49(4):399–418.

Robin Cooper and Staffan Larsson. 1999. Dialogue moves and information states. In H.C. Bunt and E. C. G. Thijsse, editors, *Proceedings of the Third International Workshop on Computational Semantics*.

Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. Match: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 376–383.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340, September. Special Issue on Spoken Language Dialogue System Engineering.

Oliver Lemon, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. The witas mult-modal dialogue system i. In *Proc. European Conf. on Speech Communication and Tech- nology*, pages 559–1562.

Colin Matheson, Massimo Poesio, and David Traum. 2000. Modelling grounding and discourse obligations using update rules. In *Proceedings of the First Conference of the North American Chapter of the Association for Computational Linguistics*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Bryan Pellom. 2001. Sonic: The university of colorado continuous speech recognizer. Technical Report TR-CSLR-2001-01, University of Colorado.

Matthew Purver. 2002. Processing unknown words in a dialogue system. In *Proceedings of the 3rd ACL SIGdial Workshop on Discourse and Dialogue*, pages 174–183. Association for Computational Linguistics, July.

Susan Robinson, Antonio Roque, Ashish Vaswani, and David Traum. 2006. Evaluation of a spoken dialogue system for military call for fire training. To Appear.

C. Rose, D. Litman, D. Bhembe, K. Forbes, S. Silliman, R. Srivastava, and K. van Lehn. 2003. A comparison of tutor and student behavior in speech versus text based tutoring.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields.

David R. Traum and Jeff Rickel. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the first International Joint conference on Autonomous Agents and Multiagent systems*, pages 766–773.

M. Walker and L. Hirschman. 2000. Evaluation for darpa communicator spoken dialogue systems.