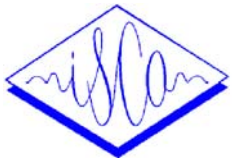

6th SIGdial Workshop on
Discourse and Dialogue
Lisbon, Portugal
September 2-3, 2005

A Corpus Collection and Annotation Framework for Learning Multimodal Clarification Strategies

Verena Rieser, Ivana Kruijff-Korbayová,
Department of Computational Linguistics
Saarland University
Saarbrücken, D-66041
{*vrieser, korbay*}*@coli.uni-sb.de*

Oliver Lemon
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, GB
olemon@inf.ed.ac.uk

Abstract

Current dialogue systems are fairly poor in generating the wide range of clarification strategies as found in human-human dialogue. The overall aim of this work is to learn when and how to best employ different types of clarification strategies in multimodal dialogue systems. This paper describes a framework for learning multimodal clarification strategies for an in-car MP3 music player dialogue system. The framework consists of three major parts. First we collect data on multimodal clarification strategies in a wizard-of-oz study. Second we extract feature in the state-action space to learn an initial policy from this data. Third we specify a reward function to refine that policy using extensions of existing evaluation schemes.

1 Introduction

Clarification strategies ensure and maintain mutual understanding in a conversation, and thus play a significant role in robust and efficient dialogue interaction. Studies of conversations between people show that there are many different types of clarification subdialogues, and that people take into account contextual as well as long term goals when deciding on their clarification strategy (Rieser and Moore, 2005). However, very few clarification strategies have been implemented in dialogue systems. The overall goal of this work is to learn when and how to best employ different types of clarification strategies in multimodal dialogue systems. This paper describes a framework for learning multimodal clarification strategies for an in-car MP3 music player dialogue system.

The methodology we are suggesting for learning a multimodal clarification policy is as follows: First we bootstrap learning from dialogue data collected using a wizard-of-oz setup. The state space and the action set are an extension of that proposed by (Georgila et al., 2005), with additional features as proposed in sections 3 and 5. An initial policy will then be generated using supervised learning techniques in the information-state update approach (ISU) to dialogue management (Lemon et al., 2005). In the ISU approach we are able to represent various kinds of dialogue features which are necessary for learning context sensitive and adaptive strategies. The strategy learnt by supervised learning reflects average human wizard behaviour. In a next step the initial learnt policy will then be refined and optimized by applying reinforcement learning (RL) to explore the policy space.

RL has been successfully applied to dialogue strategies which require complex decision making and exhaustive planning towards reaching a goal. For instance, RL has been used to optimise confirmation and initiative behaviour (Litman et al., 2000), and for deciding on the summarisation strategy for an e-mail agent (Walker, 2000). The central idea in the use of machine learning in dialogue management is to define performance functions (rewards) for combinations of (dialogue) actions and states at a particular time, with the goal of finding the policy (combinations of acts with respect to states) which maximizes total expected reward (Young, 2000).

Previously, decision theoretic methods were applied to clarification strategies, (Horvitz and Paek, 2001). Decision theoretic approaches only consider the local utility of an action. We propose that considering "delayed rewards" in RL in combination with a continuous expression of locally assigned reward signals is especially suited for clarification sub-dialogues.

In summary, in order to bootstrap an RL-based clarification strategy the following steps are required:

- Collect training data that reflects the environ-

ment (as summarised in section 2).

- Extract features in the state-action space to learn an initial policy (as listed in section 3).
- Compute a reward function to refine that policy (as described in section 4).

This paper focuses on these 3 steps. In section 5 we also describe extensions for performance modelling.

2 Data collection in a wizard-of-oz experiment

2.1 Motivation

In previous work we investigated how humans ask for clarification in task-oriented dialogue (Rieser and Moore, 2005). We were able to identify features influencing human clarification strategies (such as relation to task success, channel quality and modalities available). We now investigate how this converts to multi-modal human-machine interaction by collecting data on clarification strategies employed by multiple human wizards in a wizard-of-oz (WOZ) trial.

2.2 Goal of the experiments

In the larger context of the TALK project[1] we developed an experimental setup to gather interactions where the wizard can combine spoken and visual feedback, namely, displaying (complete or partial) results of a database search, and the user can speak or select on the screen.

One goal of the WOZ experiment was to gather data on spoken and graphical clarification strategies as employed by multiple wizards and the performance of those strategies. In particular we are interested in what medium the wizard chooses for the CR, what kind of grounding level he addresses, and what "severity" [2] he indicates. The wizards' responses were therefore not constrained by a script, but the wizard can talk freely and choose between four types of screen outputs which were automatically generated. To get realistic decisions to guide policy design the wizard only "sees what the systems sees", i.e. features which are available for decision making at system runtime.

[1]TALK (Talk and Look: Tools for Ambient Linguistic Knowledge; `www.talk-project.org`) is funded by the EU as project No. IST-507802 within the 6th Framework program.

[2]Severity describes the number of hypotheses indicated by the speaker: having no interpretation, an uncertain interpretation, or several ambiguous interpretations.

2.3 The in-car MP3 music player domain

For the MP3 music player domain the number of ambiguities is limited and can (partially) be controlled for collecting data on these kinds of phenomena. For instance an ambiguous lexical item such as a title can either be an album name, a name of an artist, or a song title. Referential ambiguity can be controlled via the number of matches in the music database. Furthermore the in-car application combines spoken and graphical interaction while the user is driving. We aim to gain initial insights regarding the difference in interaction flow under such conditions, particularly with regard to multimodality.

2.4 Experimental setup

We briefly summarise here some details of the experiments. A full description of the setup can be found in (Kruijff-Korbayová et al., 2005). The experimental setup is shown schematically in Figure 1. There are five people involved in each session of the experiment: an experiment leader (not shown), two transcribers, a user and a wizard.

The wizards play the role of an intelligent interface to an MP3 player and are given access to a database of information. Subjects are given a set of predefined tasks and are told to accomplish them by using an MP3 player with a multimodal interface. In a part of the session the users also get a primary driving task, using the *Lane Change* driving simulator (Mattes, 2003). This enabled us to collect dialogue data combining primary and secondary tasks in our experimental setup.

The wizards can speak freely and display the search results or the playlist on the screen. The users can also speak, as well as making selections on the screen. The user's utterances are immediately transcribed by a typist and also recorded. The transcription is then presented to the wizard. We did this in order to deprive the wizards of information encoded in the intonation of utterances, and in order to be able to corrupt the user input in a controlled way, simulating understanding problems at the acoustic level. The wizard's utterances are also transcribed (and recorded) and presented to the user via a speech synthesiser.

2.5 Invoking clarification behaviour

2.5.1 Method

In order to invoke different kinds of clarification behavior we introduced uncertainties on several levels, for example, multiple matches in the database, lexical ambiguities, and errors on the acoustic level.

To approximate speech recognition errors we used used a tool that "deletes" parts of the transcribed

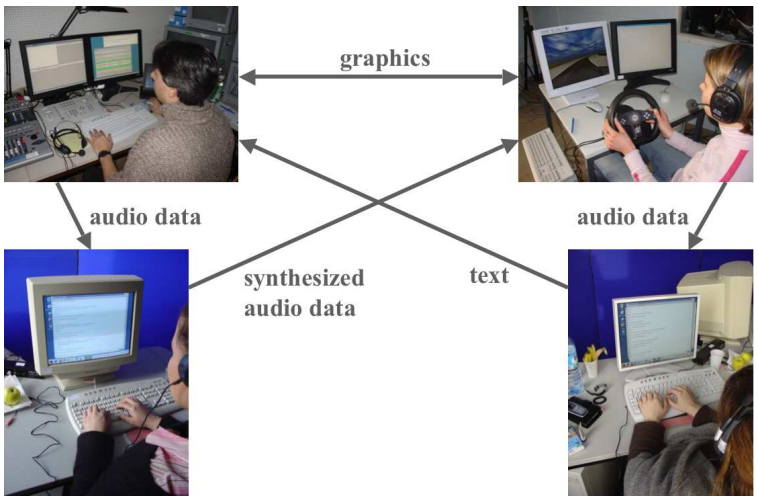

Figure 1: Multimodal Wizard-of-Oz data collection setup for an in-car music player application, using the Lane Change driving simulator. Top right: User, Top left: Wizard, Bottom: transcribers.

utterances. Due to the fact that humans are able to make sense of even heavily corrupted input, this method not only covers non understandings, but wizards also built up hypotheses, which can lead to misunderstandings. For the algorithm the word deletion rate varied: 20% of the utterances got weakly corrupted (= deletion rate of 20%) and 20% were strongly corrupted (= deletion rate of 50%). In 60% of the cases the wizard saw the transcribed speech uncorrupted. Example 1 illustrates the kind of corrupted utterances the wizard had to deal with:

(1) **uncorrupted:** Zu dieser Liste bitte Track 'Tonight' hinzufügen.
[Add track 'Tonight' to this list.]
weak: Zu dieser Liste bitte Track Tonight ....
[... track 'Tonight' to this list.]
strong: Zu dieser ... bitte Track ....
[... track 'Tonight' to this ....]

Whenever the wizard made a CR, the experiment leader invoked a questionnaire window on the screen, where the wizard had to classify his CR according to the primary source of the understanding problem. The wizards could choose between the options presented in the first two columns in table 1. Following the methods described by (Allen and Core, 1997), binary decision trees were designed to guide the classification process and training was provided.

2.5.2 Evaluation of the setup

Results: The corpus gathered with this setup comprises 1772 turns and 17076 words. Of the 774 wizard turns 10.2% are CRs, already annotated via the questionnaire window. In human-human task-oriented dialogues the frequency of CRs is about half (Rieser and Moore, 2005), indicating that the setup is suitable to elicit clarification behaviour. We expect the total number of clarifications to be even higher since the questionnaire window was sometimes not shown to the wizard. [3] The frequency of the understanding problems (as indicated by the wizard on the questionnaire window) is presented in the third column in table 1. One third of the understanding problems seem to be caused by partial non-

Level	Severity	Frequency
Contact	unsure	10.2%
String level	partially unclear	33.3%
	totally unclear	13.0%
Relation DB	none	14.5%
	unsure	—
	two or more	1.5%
Search results	none	1.5%
	conflicting	2.9%
	too many	15.9%

Table 1: Options for describing understanding problems as displayed to the wizard

[3] There were several reasons for not showing the questionnaire window. Either the experiment leader deliberately chose not to disturb the wizard, or was undecided about the request being about clarification, or wasn't available at that moment.

understanding on the string level. The second two most frequent understanding problems are "lexical" interpretation errors, i.e. the wizard did not know what to search for, and reference problems caused by too many matches in the database. 7.25% CRs were indicated as "other" and for the same number of times the wizard did not react to the questionnaire window.

CRs in WOZ vs. human-human dialogues: In previous work we annotated CRs in the human-human travel reservation dialogues held via telephone, available as part of the CMU COMMMUNICATOR Corpus (Bennett and Rudnicky, 2002) as described in (Rieser and Moore, 2005). In both studies CRs frequently address the acoustic level caused by bad channel quality (31.0% for COMMMUNICATOR and 46.3% for WOZ). For the WOZ setup acoustic problems (set equal to string level problems) are worse than for human-human dialogues. For human-human dialogues only 0.03% are complete acoustic non-understandings whereas, due to our word deletion algorithm, for the WOZ study 13.04% were complete acoustic non-understandings. Partial acoustic understanding problems where about equal (30.97% for COMMMUNICATOR and 33.3% for WOZ).

In human-human dialogue lexical problems are rare. In the WOZ study lexical problems were the third most frequent, reflecting unknown words in dialogue systems.

Finally, reference problems are almost twice as likely for the human-human dialogues as for the WOZ study (39.8% vs. 20.34%). In the music domain correct values for every slot are not as critical as for travel booking. Examples from the corpus show that wizards often choose not to clarify an item which has multiple matches in the DB but would chose a default value (i.e. based on frequency or popularity). In future work we will investigate the performance of this strategy in comparison to asking direct clarification requests.

In sum, we can conclude that this WOZ setup successfully invokes a high number of clarification requests while simulating the kinds of errors found in spoken dialogue systems.

2.5.3 Limitations of the method

Although showing some progress in simulating understanding problems as they occur in dialogue systems, this method has several obvious limitations.

- The system's problems on the acoustic level caused by imperfect speech recognition are more severe than simulated by our word-deletion tool. Parsing strategies employed by dialogue systems are less robust than the ones by human wizards.
- The overall setup caused a time delay which had a negative influence on user satisfaction as well as on the clarification strategy. In the debriefing session all the wizards reported that they adapted their behaviour. They asked shorter questions and sometimes dispensed some requests completely.
- The problem sources selected on the questionnaire window cannot be considered completely reliable. Some of the wizards reported that the categories were unclear to them or were too general. Furthermore the pop-up window was sometimes distracting them from their primary search task.

For all these reasons a clarification strategy learnt from human wizards via supervised methods will only be sub-optimal for dialogue systems. Therefore we will need to apply RL to optimise the clarification policies.

3 Extracting context features for learning

For applying RL we need to define an initial policy. In most of the work to date the initial strategy is handcrafted. However, we want to reflect strategies used by human wizards. Elsewhere we show that mean user satisfaction is fairly high across wizards (Kruijff-Korbayová et al., 2005), meaning that we can employ the wizards' strategies as a baseline. We therefore use the data collected in the WOZ trail as training data to bootstrap a policy using supervised learning.

In this section we define features for annotating the collected data with features of the state-action space used for learning an initial strategy. For the state space we will be following an automatic annotation method introduced by (Georgila et al., 2005), and many of the features are already automatically logged by the experimental system.

The data logged per dialogue turn is:

- manually transcribed user speech (online and offline)
- corrupted user speech
- transcribed wizard's speech (online and offline)
- wizard's database query
- database query results
- graphical options as presented to the wizard

- graphical option chosen by the wizard for display
- user clicks
- the CR and primary problem as chosen by wizard on the questionnaire window

Other features are inferred/computed from the logs.

3.1 Annotation scheme for CRs

Based on the classification scheme of (Rodríguez and Schlangen, 2004) we developed a four dimensional scheme to annotate functions of CRs and the modality used to present those functions. We are using this annotation scheme to discover the action set, i.e. the clarification requests, through annotation.

- **Severity:** indicates how much was understood by analysing what kind of answer the CR initiator requests from the addressee. "Severity" can take the values: `content repetition`, `content confirmation`, `content disambiguation`.
- **Source:** primary source for interpretation uncertainty as indicated by the CR initiator, taking the possible values: `acoustic`, `lexical`, `syntactic`, `reference`, `intuition`, etc.
- **Extent:** the CR initiator points out one part of the utterance, taking the values: `whole`, `part`.
- **Modality:** modality used, taking the values: `speech`, `graphics`, `both`.

3.2 Annotation principles for ISU systems

A state in our system is a dialogue information state as defined in (Lemon et al., 2005). Following (Georgila et al., 2005) we divide the types of information represented in the dialogue information state into 5 main levels: *dialogue-level, low-level, task-level, history-level*, and *reward level*. (Georgila et al., 2005) divide the logging and annotations required into information about utterances, and information about states. In addition to state and utterance, we defined features reflecting the application environment, e.g. whether the user is driving, the number of matches from the database query, how many display templates were generated (if the database query returned too many matches only the text option was generated), which one was chosen by the wizard etc. Note that multimodal features also need to be annotated. We describe these in section 5 below.

One problem of applying RL in the ISU approach is the large state space. For initial policy learning we apply a supervised learning technique, namely maximum entropy modelling, which learns how to set feature weights automatically. This will provide us valuable information what feature set performs best in order to reduce the sate space.

4 Performance modelling with DATE and PROMISE in PARADISE

For applying RL to dialogue design the clarification problem is reformulated in terms of a Markov Decision Process (MDP). A MDP is a collection of four elements: the set of states S, the set of actions A, the transition probabilities T, and a set of rewards R, (Sutton and Barto, 1998). How to define S and A through annotation is described in the previous section. The n-best results from the probability distribution we get as an output from the maximum entropy model defines T for each state-action pair. Some values for R are already logged or elicited via user questionnaires. Introducing a reward function allows us to create or refine the policy using RL.

As the strategies represent in our corpus are only sub-optimal, RL needs to explore unobserved state spaces. Such policy exploration is only feasible with simulated dialogues generated through interaction with a simulated user. Associated work on user simulation can be found elsewhere (Schatzmann et al., 2005). In this work we concentrate on producing better online reward measures for unobserved states.

4.1 RL and performance modelling

In RL, the objective of the system is to maximise the reward it gets for the action choices during the course of the dialogue. Rewards are defined to reflect how well a dialogue went, so by maximising the total expected reward the system optimises the quality of the dialogue. The difficulty is that, at any point in the dialogue, the system cannot be sure what will happen in the remainder of the dialogue, and thus cannot be sure what effect its actions will have on the total reward at the end of the dialogue. Thus the system must choose an action based on the average reward it has observed earlier when it has performed that action in states similar to the current one. This average is the expected future reward. The core component of any RL system is the estimation of the expected future reward (the Q-function). Given a state and an action that could be taken in that state, the Q-function tells us what total reward, on average, we can expect between taking that action and the end of the dialogue. Once we have this function,

the optimal dialogue management policy reduces to simply choosing the action (a) which maximises the expected future reward ($E[]$) for the current state (s_i). The maximised Q-function is what we call a "value function" (V-function).

$$Q(s_i, a) \approx E[\sum_{j \geq i} r(d, j) | s_i, a] \quad (2)$$

System designers have to define a mapping $r(d, i)$ from a dialogue d to a position in that dialogue i to a reward value. Previous work applied simple reward functions such as task completion or some measure of user effort such as elapsed time or number of user turns. But it is in general agreed that dialogue system design should aim to optimise user satisfaction. For RL we need a definition of user satisfaction that can be calculated online.

We now describe a combination of three schemes used to model user satisfaction for dialogue systems. The PARADISE framework allows us to combine multiple evaluation matrices to automatically predict user satisfaction. DATE is a dialogue tagging scheme for evaluation which refines cost measures, and PROMISE defines a framework to compute task success more dynamically.

4.2 The PARADISE framework

(Walker, 2000) successfully applied the PARADISE evaluation framework to learn a performance function (reward) used in reinforcement learning. The framework posits that user satisfaction is the overall objective to be maximised and that task success and various interaction costs can be used as predictors of user satisfaction. Common measures used in PARADISE are dialogue efficiency metrics (such as elapsed time, system turns), dialogue quality metrics (such as the mean recognition score), task success metrics and other factors that contribute to user satisfaction (such as task ease, interaction pace, future use).

4.3 The DATE scheme

Studies have shown that a more fine-grained model than the one used by PARADISE is necessary to evaluate dialogue quality. In a WOZ study by (Williams and Young, 2004) user satisfaction was not correlated with turn duration at all. In a study by (Walker et al., 2001) turn duration is even positively correlated with user satisfaction. This is because, for certain dialogues, turn duration seems to predict task success. The DATE scheme takes into account such relations by allowing multiple views on one turn, namely the *conversational domain*, *task* and *sub-task* level, and the *speech act* performed. We extend the DATE scheme to capture user actions, as described in (Georgila et al., 2005), and for multimodal actions as proposed in section 5. Except for speech acts we have all this information in our system logfiles.

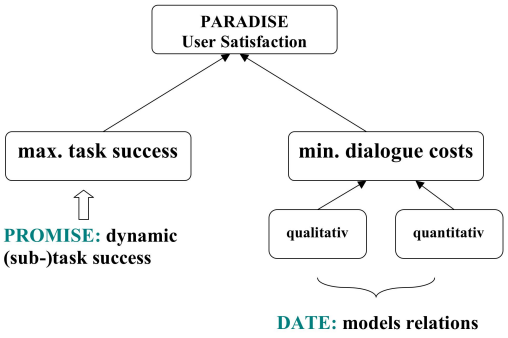

Figure 2: Extended PARADISE framework for estimating user satisfaction

4.4 Task-success in PROMISE

A further extension to PARADISE is the PROMISE framework (Beringer et al., 2002). It suggests *information bits* to deal with non-directed task definitions and the resulting, potentially uncompleted, tasks in multimodal dialogues. In PARADISE overall task-success in defined by an AVM-style definition, being either 1 or 0. This is reasonable for a task like train booking (for which PARADISE was developed) but for other domains a more flexible definition of sub-task success is necessary. In PROMISE the task descriptions were quite vague ("Plan an evening watching TV") and can be accomplished by providing different kinds of information (e.g. a film can be named by title and channel or by title and time or just by clicking on the item on a screen). The same is true for the music domain, so we adopt the PROMISE framework in our experiments.

In sum, PARADISE allows to estimate user satisfaction automatically by combining features indicating task success and dialogue costs. The aim of maximising task-success directs learning towards robust clarification strategies, while the aim of minimising costs directs it towards efficient clarification stratgegies. PROMISE refines the task success measures by accounting for alternative ways to accomplish a (sub-)task. DATE refines the cost measures by accounting for relations between quantitative and qualitative features. Figure 2 presents these relations

schematically.

5 Extensions to DATE and PROMISE

To automatically estimate user satisfaction for the data collected in the WOZ study we need to account for costs caused by multimodal speech acts and for undirected task descriptions. In this section we describe extensions to DATE and PROMISE that will allow us to calculate user satisfaction for multimodal strategies at system runtime.

5.1 Multimodality in DATE

Annotating the collected data with the DATE scheme requires that we include another dimension capturing multimodality. Adding a multimodal dimension to the DATE scheme allows us to capture features which are said to be typical for multimodal interaction like providing different information *simultaneously* and providing *redundant* information across modalities, by relating multimodal "speech" acts to speaker turns. Consider the example annotated corpus extract in table 3. In turns 2 and 3 the wizard performs two acts within the same dialogue turn but in different media. By relating this information to the task layer we can measure how multi-tasking speeds up task completion. In turns 4 and 5 the user performs the same speech act twice in different media. For performance modeling this phenomenon can be related to task precision.

5.2 Ambiguity in PROMISE

In the PROMISE scheme *information bits* can be compared to different sets of slot-value pairs which need to be filled to accomplish a task. In our domain we face two challenges. First, it is not clear how many slots are relevant for a task to be completed. As some of our task descriptions are quite vague it is the user's goal which defines task success. Second, values specified by the user can have several matches in the database, i.e. they are ambiguous.

Consider the following example of a task description:

> *Your little brother likes to listen to heavy metal music. You want to build him a playlist including three metal songs. Make sure you have "Enter Sandman" on the playlist! Save the playlist under the name "heavy guys".*

For this task (`makePlaylist`) there are 7 sub-tasks to be accomplished, `search(item1)`, `search(item2)`, `search(item3)`, `playlist(name)`, `add(item1,name)`, `add(item2,name)`, `add(item3,name)`. With respect to PROMISE these sub-tasks can be accomplished by providing different sets of information bits. For example for `search(item1)`, item1 can be described by a title, or an album and the track number, or the track number of a displayed list, or a click on an item on that list. One of the items is constraint by a song title. The dialogue designer would specify the information bits needed as follows: `item1`=[`title`] ∨ [`album,track`], `item2`= [`title:` " `Enter Sandman`"]... In our domain all of those values can be ambiguous. For example searching for the song "Enter Sandman" will return several matches in the DB. The item can only be defined by providing another information bit, like album or artist. Meaning that we have an interplay between information bits and their values. For ambiguous items an initially defined information set is not able anymore to precisely describe task success as we need another information bit to identify the item. On the other hand we do not know anything about the user's goals. Users might not want to be as precise and accept "default" values for some slots as results from the WOZ study do show.

To handle this dilemma we use a localised reward measure for every instantiated information bit, i.e. once a slot which is relevant for the task gets filled and confirmed we assign positive rewards. For computing the final task success we use a flexible backing off algorithm. Every time an ambiguity is detected the information bit set which was currently instantiated (i.e. the one of all possible alternative sets whose information bits is most similar to the currently filled slots) gets extended with another constraint and the new set is added to the set of alternative information bit sets. At the end of the dialogue we "back off" to the maximal information bit set which got instantiated by the feature-value pairs provided by the user, considering this set as the "user's goal". Figure 3 shows the pseudo-code for computing an extended definition of task success.

5.3 Implications for the reward measure

Applying estimated user satisfaction as defined by PARADISE as a reward function will only provide us with a dialogue-final reward measure which is less informative and more costly for learning. Some RL-based systems use the weighted sum of per turn penalties and some measure of task success as their reward function, reflecting the idea of the PARADISE framework but acting more locally.

Given the constraints discussed above the definition of a reward function is not straightforward. The local and final task success measures calculated in the

```
U is user input string
DB is number of matches in the database
Initialize:
      task = makePlaylist
      makePlaylist = subtask(item1) ∧ ... ∧ subtask(itemN)
      item1, ..., item N = alternativeSetList
      alternativeSetList =infoSet1 ∨ infoSet2 ∨ ... ∨ infoSetN
      infoSet1, infoSet2, ..., infoSetN = infoBit1 ∧ infoBit2 ∧ infoBitN
For every U:
     value = Parse(U)
     If (DB != 0):
               newSet = currentSet.add(infoBit)
               alternativeSetList.add(newSet)
For every infoSet in alternativeSetList:
      try to instantiate infoSet
      currentUserGoal = infoSet instatiated
```

Figure 3: Pseudo-code for update task success

modified PROMISE framework overcome the difficulty of using purely delayed rewards. However the relationship between speech acts and task completion as modelled by DATE also needs to be reflected in the reward measure. A way to communicate complex information in RL is to apply *policy shaping*. The idea behind shaping is to augment the underlying reward structure with more informative local rewards, represented by a shaping function F which is a representation of a bias reflecting prior domain knowledge (Laud and DeJong, 2002). The result is faster learning at the cost of more uniform exploration across policies.

For our task we still lack knowledge of how the relation between multimodal speech acts and cost features is to be defined. *Dynamic shaping* allows us to specify a shaping function even if prior knowledge is uncertain. The parameters of F are adjusted through initial observation of world interactions via a mediating explanation, i.e. the specified relationship. RL then proceeds as before.

6 Discussion and future work

To this point we have not addressed the problem of how to account for more user-centred qualitative features in defining the reward function, nor how to account for the additional cognitive load imposed by the driving task. We hope to further improve the predictive power of our model of user satisfaction by adding user "emotions", which we conjecture are continuous expressions of reward. By giving immediate reward/punishment for some dialogue actions we also hope to learn a clarification strategy that will react to user frustration and stress more quickly. Especially for dialogues in the in-car domain this will be valuable information. Subjects reported in the debriefing session that some multimodal feedback strategies were imposing a high cognitive load when driving.

For example, we initially propose annotating simple user expressions of positive and negative feedback, such as "great", "thank you", "damn" etc. and use these as immediate reward signals. We plan to test this hypothesis on COMMUNICATOR data, which is already annotated with task completion reward signals (Lemon et al., 2005; Georgila et al., 2005).

7 Conclusion

We have presented a data collection and annotation framework to collect a corpus suitable for reinforcement learning of multimodal clarification strategies. We described a wizard-of-oz setup used to gather the data for learning, for an in-car music player dialogue system where driving is the primary task, and dialogue is secondary. We explained the constraints that reinforcement learning places on the corpus and its annotation, and we briefly explained how to combine reinforcement learning methods with the information state update approach to dialogue management (Lemon et al., 2005).

To model user satisfaction we proposed an extended metric of dialogue quality and task-success based on two existing schemes, namely DATE, (Walker and Passoneau, 2001), and PROMISE, (Beringer et al., 2002). We also proposed extensions to the DATE scheme to cover multimodal dialogue acts. We argued that the more flexible definition of task success in PROMISE is needed to account for non-directed task definitions and ambiguity. Finally we discussed implications for formulating the reward function using policy shaping and provided an out-

look on "emotion" tagging for learning clarification strategies.

Acknowledgements

This work is partially supported by the TALK project (Talk and Look: Tools for Ambient Linguistic Knowledge; `www.talk-project.org`).

References

[Allen and Core1997] James Allen and Mark Core. 1997. Draft of DAMSL: Dialog act markup in several layers.

[Bennett and Rudnicky2002] Christina L. Bennett and Alexander I. Rudnicky. 2002. The Carnegie Mellon Communicator Corpus. In *Proceedings of the International Conference of Spoken Language Processing (ICSLP02)*.

[Beringer et al.2002] Nicole Beringer, Ute Kartal, Katerina Louka, Florian Schiel, and Uli Türk. 2002. PROMISE: A procedure for multimodal interactive system evaluation. In *Proceedings of the Workshop Multimodal Resources and Multimodal Systems Evaluation*.

[Georgila et al.2005] Kallirroi Georgila, Oliver Lemon, and James Henderson. 2005. Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations. In *Proceedings of DIALOR, 9th Workshop on the Semantics and Pragmatics of Dialogue*.

[Horvitz and Paek2001] Eric Horvitz and Tim Paek. 2001. Harnessing models of users' goals to mediate clarification dialog in spoken language systems. In *Proceedings of the 8th International Conference on User Modeling*.

[Kruijff-Korbayová et al.2005] Ivana Kruijff-Korbayová, Nate Blaylock, Ciprian Gerstenberger, Verena Rieser, Tilman Becker, Michael Kaisser, Peter Poller, and Jan Schehl. 2005. An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system. In *10th European Workshop on Natural Language Generation*.

[Laud and DeJong2002] Adam Laud and Gerald DeJong. 2002. Reinforcement learning and shaping: Encouraging intended behaviour. In *Proceedings of the 19th International Conference on Machine Learning*.

[Lemon et al.2005] Oliver Lemon, Kallirroi Georgila, James Henderson, Malte Gabsdil, Ivan Meza-Ruiz, and Steve Young. 2005. D4.1: Integration of learning and adaptivity with the ISU approach. Technical report, TALK Project.

[Litman et al.2000] Diane Litman, Micheal Kearns, Satinder Singh, and Marylin Walker. 2000. Automatic optimization of dialogue management. In *Proceedings of COLING*.

[Mattes2003] Stefan Mattes. 2003. The lane-change-task as a tool for driver distraction evaluation. In *Proceedings of IGfA*.

[Rieser and Moore2005] Verena Rieser and Johanna Moore. 2005. Implications for Generating Clarification Requests in Task-oriented Dialogues. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, ACL*.

[Rodríguez and Schlangen2004] Kepa Rodríguez and David Schlangen. 2004. Form, Intonation and Function of Clarification Requests in German Task-oriented Spoken Dialogues. In *Proceedings of the CATALOG, 8th Workshop on Formal Semantics and Dialogue*.

[Schatzmann et al.2005] Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. uantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

[Sutton and Barto1998] Richard S. Sutton and Anrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. The MIT Press.

[Walker and Passoneau2001] Marylin Walker and Rebecca Passoneau. 2001. DATE: A dialogue act tagging scheme for evaluation. In *Human Language Technology Conference*.

[Walker et al.2001] Marilyn Walker, Rebecca Passonneau, and Julie Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. *Meeting of the Association of Computational Linguistics*.

[Walker2000] Marylin Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, (12):387–416.

[Williams and Young2004] Jason Williams and Steve Young. 2004. Characterizing task-oriented dialog using a simulated ASR channel. In *Proceedings of ICSLP*.

[Young2000] Steve Young. 2000. Probabilistic methods in spoken dialogue systems. *Philosophical Trans Royal Society*.

Dialogue level	**Low level**	**Task level**	**History level**	**Reward level**
• dialogue act • corrupted user string • key word deletion rate (KDR) • wizard output string • modality • salient NPs • salient VPs • user driving	• time delay • DB query • graphical templates generated • graphical templates displayed • user clicks	• task/ subtask-type • DB matches • number of DB matches • changes to filled slot values • user goals	• cumulative filled slot values • number of clarifications • last N user dialogue acts • last M system dialogue acts • number of abandoned (sub)tasks • dialogue duration • number of turns • average/min/max KDR • last X system modalities • last Y user modalities	• task completion (actual and perceived) • task satisfaction • dialogue duration • number of turns • user satisfaction • Paradise evaluation metrics

Table 2: ISU context features + reward annotations

ID	Utterance	Speaker	**Modality**	Speech act	Task	Domain
1	Please play "Nevermind".	user	speech	request	play song	about task
2	Does this list contain the song?	wizard	speech	request info	play song	about task
3	[shows list with 20 DB matches]	wizard	graphic	present info	play song	about task
4	Yes. It's number 4.	user	speech	provide info	play song	about task
5	[selects item 4]	user	graphic	provide info	play song	about task

Table 3: Example corpus extract showing extended DATE annotation capturing multimodality