DIALOGUE STRATEGIES GUIDING USERS TO THEIR COMMUNICATIVE GOALS

Matthias Denecke
denecke@mcspeech.com
Multicom Research
1900 Murray Avenue
Pittsburgh, PA 15217,USA

Alex Waibel
ahw@cs.cmu.edu
Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract

Much work has been done in dialogue modeling for Human - Computer Interaction. Problems arise in situations where disambiguation of highly ambiguous data base output is necessary. We propose to model the task rather than the dialogue itself. Furthermore, we propose underspecified representations to represent relevant data and to serve as a base for generating clarification questions that guide the user efficiently to arrive at his communicative goal. In this paper, we establish a connection between underspecified representations as representations of disjunctions and clarification questions. Our approach to clarifying dialogues differs from other approaches in that the form of the clarification dialogues is entirely determined by the domain modeling and by the underspecified representations.

1 Introduction

In spoken dialogue systems, the need for clarification questions arises in situations in which information is missing (e.g. due to partial interpretation in the presence of recognition errors) or in situations in which interpretations of the speech acts are ambiguous (e.g. due to not sufficiently specified database requests). A straightforward approach to circumvent these problems is to query complementary information until the required degree of specificity is reached. A frequently applied strategy in frame-based dialogue systems is to associate a predetermined question with a slot and to ask the question every time the filler of the slot is missing. However, the straightforward approach has some inherent problems. Among the open questions are: What information can disambiguate an ambiguous representation most efficiently, especially if there are several possible questions that may be asked? If information is missing, how can one provide the user with all options available at this point of the dialogue?

To overcome these problems, we propose a departure from the model-based approach to dialogue processing in favor of an information-based approach [Denecke, 1997]. By information-based approach, we understand that the specificity of the information available at any given point in the dialogue, comprising results from database requests, determine the actions to undertaken by the dialogue system. We propose to model the domain of the dialogue, as well as the services the dialogue system offers, in a type hierarchy. We describe how the information provided by the type hierarchy can be exploited to transform representations with missing information into underspecified representations that we use as the base for clarification questions. Moreover, we propose to use underspecified typed feature structures to represent sets of objects. Since underspecified feature structures leave disjunctions unresolved, the dialogue strategy reduces to disambiguating underspecified feature structures both when relevant information is missing and database requests are not sufficiently specified. We show how underspecified feature structures can be exploited to generate clarification dialogues.

We assume that a dialogue system offers a limited set of services and that these services are among the possible communicative goals of the user. We propose to specify communicative goals by means of lower informational bounds, i.e. by typed feature structures that subsume possible communicative goals. Thus, a clarification dialogue can be seen as a sequence of questions whose answers are incorporated in a monotonic way into the initially deficient representation to meet the lower bound of the communicative goal. The described representations are the only input to the algorithms that decide which questions to ask and which information to convey in the question. Consequently, this approach is entirely data-driven and is only dependent on the situation.

The paper is organized as follows. In section 2, we present underspecified typed feature structures. In section 3, we describe how to specify a communicative goal. In section 4, we describe the way relevant information is determined to generate clarification questions. Section 5 summarizes the paper.

2 Representations

We encode domain knowledge in a type hierarchy and use typed feature structures over this type hierarchy to represent objects in the domain. Underspecified feature structures represent sets of typed feature structures. Moreover, generalizations of a set of typed feature structures represent the similarities of all feature structures in the set.

2.1 Domain Modeling

We chose to represent descriptions of objects using typed feature structures [Carpenter, 1992]. The types are ordered in a so-called *type hierarchy* which represents domain-specific terminological knowledge using IS-A and IS-PART-OF relations. We restrict the type hierarchy to be a tree and assign probabilities to the edges of the tree expressing the degree of evidence that if an object is of type θ it is also of type θ', where θ' is subsumed by θ. Figure 1 shows a part of type hierarchy used in an interactive map application.

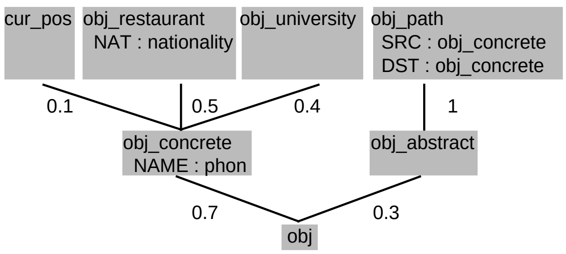

Figure 1: A part of the type hierarchy and its appropriateness conditions used in the map application. The least specific type is at the bottom of the tree. Information increases from the bottom to the top.

By imposing lower bounds on the feature values for all possible feature for a given type, type inference [Carpenter, 1992] is possible.

2.2 Underspecified Representations

The underspecified feature structures are a generalization of the typed feature structures allowing to represent descriptions of more than one object. They are such that the similarities and differences of objects are transparent. An example of an underspecified feature structure is given in figure 2.

An underspecified feature structure is a compact representation of the F_i than their disjuncts in that it represents common information only once. This is a crucial property when generating clarification questions.

2.3 The Generalization

The generalization of a set of feature structures $F_1, \ldots, F_n$, as dual to its unification, is defined as the least specific feature structure F that subsumes all the F_i. Since the underspecified feature structure $F_\sim$ representing the F_i factores out all common information of every subset of $\{F_1, \ldots, F_n\}$, the generalization of every subset is represented in $F_\sim$. Figure 3 shows two generalizations represented by the underspecified feature structure in figure 2. We use the generalization to detect an increase of information when disambiguating underspecified feature structures.

3 Specifying the Communicative Goal

We adopt the hypothesis that a dialogue system is intended to perform a limited set of parametrized actions and that the communicative goal of the user is to fully specify one of these actions. Furthermore, the actions to be performed establish some inherent lower bounds of specificity on their parameters. For example, the destination address of a path to be calculated by a map-based application has to be unique while the house number might be missing in which case the system would take the first intersection encountered on the street. On the other hand, an object to be displayed on the map does not have to be uniquely specified (a set of objects might be displayed as well) but both street name and house number have to be provided in order to determine the location of the objects on the map.

$$
\begin{bmatrix}
\textit{obj\_restaurant*} \left\{ \begin{bmatrix} \textit{obj\_restaurant} & \\ \textsc{spec} & \textit{spec} \left\{ \begin{array}{l} \textit{" patio " (1,2)} \\ \textit{" live\_music " (4)} \end{array} \right\} \end{bmatrix} \right\} & \\
\textsc{name} & \textit{string*} \left\{ \begin{array}{l} \textit{" Abruzzi's Restaurant "(1)} \\ \textit{" Antonini Ristorante "(2)} \\ \textit{" Cozumel "(3)} \\ \textit{" Mad Mex "(4,5)} \\ \textit{" Mediterranean Grill "(6)} \end{array} \right\} \\
\textsc{addr} & \begin{bmatrix} \textit{address} & \\ \textsc{str-name} & \textit{string*} \left\{ \begin{array}{l} \textit{" forbes ave" (1,3)} \\ \textit{" jane st " (2,4,6)} \\ \textit{" atwood st " (5)} \end{array} \right\} \\ \textsc{str-num} & \textit{int*} \left\{ \begin{array}{l} \textit{52(1)} \\ \textit{370(2)} \\ \textit{2700(3)} \\ \textit{5505(4)} \\ \textit{5824(5)} \\ \textit{6400(6)} \end{array} \right\} \end{bmatrix} \\
\textsc{nationality} & \textit{nationality*} \left\{ \begin{array}{l} \textit{mexican (3,4,5)} \\ \textit{greek (6)} \\ \textit{italian (1,2)} \end{array} \right\}
\end{bmatrix}
$$

Figure 2: An underspecified feature structure representing six restaurants. Two of the restaurants have a patio, one offers live music.

Given these assumptions, a *communicative goal* can be specified by a typed feature structure in which the feature values impose lower bounds on the parameters of the given action. As such, a communicative goal is a feature structure that subsumes any well-defined representation of this particular action. We define a *dialogue strategy* as a sequence of actions, undertaken by the dialogue system, whose end it is to generate a feature structure that meets the informational lower bound of exactly one communicative goal.[1]

An example of the specification of the communicative goal to reserve a hotel room is given in figure 4. The types *obj_hotel* and *date* are not atomic; the specification means that the values of each feature of these types have to be uniquely specified after a database request took place and possibly missing information is filled in. This allows for requests like `I need a room in the cheapest hotel from next Tuesday on` which triggers a database request to determine the cheapest hotel and the explicit representation of the date.

4 Generating Questions

In almost all cases, asking a clarification question can be seen as the user selecting one option out of a list of

[1]This includes the abortion of an action, provided that the atomic feature structure [*speechact_nullaction*] is among the communicative goals. However, this would require non monotone updates that are handled only rudimentarily by the current implementation of the system.

$$(a)\quad \left[\begin{array}{ll} \textit{obj\_restaurant} & \\ \text{NAME} & \textit{" Mad Mex "} \\ \text{ADDR} & \left[\begin{array}{ll} \textit{address} & \\ \text{STR-NAME} & \textit{string} \\ \text{STR-NUM} & \textit{int} \end{array}\right] \\ \text{NATIONALITY} & \textit{mexican} \end{array}\right]$$

$$(b)\quad \left[\begin{array}{ll} \textit{obj\_restaurant} & \\ \text{NAME} & \textit{string} \\ \text{ADDR} & \left[\begin{array}{ll} \textit{address} & \\ \text{STR-NAME} & \textit{string} \\ \text{STR-NUM} & \textit{int} \end{array}\right] \\ \text{SPECIALTY} & \textit{patio} \\ \text{NATIONALITY} & \textit{italian} \end{array}\right]$$

Figure 3: Two generalizations over subsets of the six restaurants: (a) shows the generalization over the restaurants carrying the name "Mad Mex", while (b) shows the generalization over the restaurants having a patio. The generalizations are deductible from the underspecified representations shown in figure 2 when restricting the represented feature structures to $\{4,5\}$ and $\{1,2\}$ respectively.

$$\left[\begin{array}{ll} \textit{speechact\_reservehotelroom} & \\ \text{OBJECT} & \textit{obj\_hotel} \\ \text{ARRIVAL\_DATE} & \textit{date} \\ \text{DEPARTURE\_DATE} & \textit{date} \\ \text{NUM\_PERSONS} & \textit{int} \\ \text{NUM\_BEDS} & \textit{int} \\ \text{BED\_SIZE} & \textit{bedsize} \end{array}\right]$$

Figure 4: The specification of a communicative goal

several possible options. The possible options are then conveyed to the user who is expected to provide information to disambiguate the underspecified representations.

4.1 Determining Options for missing Information

If relevant information is missing in the request or has been skipped due to recognition errors, lower bounds on the missing information are given by the underspecified representation of all communicative goals that are compatible with the information currently present. As an example, consider a map application, in which the partial representation of a request states that the user talks about a hotel while the representation of the action is missing. In the application, the path to a hotel can be calculated, a description of the hotel can be shown or a reservation can be made. The underspecification of all compatible representations of communicative goals is shown in figure 5.

In this way, missing information is transformed to disjunctively specified representations whose disjunctions are resolved using clarification questions. Moreover, the information in the disjuncts lets the system guide the user and reduces the incorporation of complementary information to disambiguation.

4.2 Determining the Form of the Question

The goal of a clarification question is to obtain information to disambiguate a representation. Which information helps to disambiguate an underspecified feature

$$\left[\begin{array}{ll} \textit{speechact*}\left\{\begin{array}{l} \textit{speechact\_reservehotelroom} \\ \textit{speechact\_printinformation} \\ \textit{speechact\_showpathtoobj} \end{array}\right\} & \\ \text{OBJECT} & \left[\begin{array}{ll} \textit{obj\_hotel} & \\ \text{NAME} & \textit{" William Penn Hotel "} \end{array}\right] \end{array}\right]$$

Figure 5: An underspecified feature structure representing possible actions to be performed on hotels

structure is determined by the underspecified representation itself: only information that is not in the generalization of a set of feature structures can be used to distinguish between these feature structures. For example, to distinguish between the two Mexican restaurants called "Mad Mex", one would have to ask for the street name. Since the differences of the objects are transparent in underspecified feature structures, they serve as a point of departure for determining the relevant information.

We chose a feature path in the underspecified structure such that its value is not uniquely determined. In figure 2, one possible feature path is NATIONALITY. In this example, the form of the question will be to let the user choose between restaurants serving Mexican, Greek or Italian food. Each of the optional types, if unified with the value of the path will disambiguate this feature path and decrease the ambiguity of the underspecified representation.

If the degree of ambiguity is relatively low, one could determine information that disambiguates the underspecified feature structure completely at once. Suppose the user chooses Mexican restaurants in our restaurant example (see figure 2) and may now choose between the remaining three Mexican restaurants. To generate unique descriptions of all three of the restaurants, we re-iterate the path selection until the information provided by the path values disambiguates the structure completely. For the restaurant called "Mad Mex", we have to choose a feature path that distinguishes between the two restaurants. In both cases, resulting from this process are feature structures representing each of the options.

	Example 1	Example 2
Disj 1	$\left[\begin{array}{ll} \textit{obj\_restaurant} & \\ \text{NAT} & \textit{italian} \end{array}\right]$	$\left[\begin{array}{ll} \textit{obj} & \\ \text{NAME} & \textit{" mad mex"} \\ \text{ADDR} & \left[\begin{array}{ll} \textit{addr} & \\ \text{STR-NAME} & \textit{"atwood"} \end{array}\right] \end{array}\right]$
Disj 2	$\left[\begin{array}{ll} \textit{obj\_restaurant} & \\ \text{NAT} & \textit{greek} \end{array}\right]$	$\left[\begin{array}{ll} \textit{obj} & \\ \text{NAME} & \textit{" mad mex"} \\ \text{ADDR} & \left[\begin{array}{ll} \textit{addr} & \\ \text{STR-NAME} & \textit{"jane"} \end{array}\right] \end{array}\right]$
Disj 3	$\left[\begin{array}{ll} \textit{obj\_restaurant} & \\ \text{NAT} & \textit{mexican} \end{array}\right]$	$\left[\begin{array}{ll} \textit{obj} & \\ \text{NAME} & \textit{" cozumel "} \end{array}\right]$

Figure 6: Example 1 shows the information chosen to disambiguate the underspecified feature structure shown in figure 2. Example 2 shows the information that disambiguates the structure in figure 2 completely after it gets unified with the third disjunct from example 1.

As a further refinement, if one reading is strongly preferred over all other readings, a yes-no question can be generated in which only confirmation for the preferred reading is asked for. The probabilities along the IS-A links in the type hierarchy are used to determine if there is a strongly preferred reading.

There are some degrees of freedom when choosing a feature path. We assume that the probabilities of the values correspond to those stored in the type hierarchy. Thus, it is possible to calculate the entropy of a feature path defined as the information that is necessary to disambiguate the path. To choose a feature path, we determine the set of all paths with maximum entropy. We assume furthermore that it is preferable to convey information stored in values of shorter feature paths. A longer feature path describes a more detailed object than a shorter one, since the features are IS-PART-OF relations. If the set of paths with maximum entropy contains more than one path, we choose the shortest feature path.

The previous step yields a set of feature structures, each of which represents one possible option. If there are few options (say, less than five), each of the options is transformed to text. If there are five options or more, the lower bound of the types of the options is calculated and mapped to a string.

The fact that an underspecified feature structures F explicitly represent the generalization of all feature structures $F_1, \ldots, F_n$ represented in F can be made use of to detect that an option initially left to the user is no longer available. Consider the following representation of three hotel rooms. If the user decides to take the room with the

$$\begin{bmatrix} \textit{obj\_hotelroom} & \\ \text{BEDSIZE} & \textit{bedsize*} \begin{Bmatrix} \textit{kingsize(1,2)} \\ \textit{double(3)} \end{Bmatrix} \\ \text{SMOKINGSTATE} & \textit{smokingstate*} \begin{Bmatrix} \textit{smoking(1,3)} \\ \textit{non-smoking(2)} \end{Bmatrix} \end{bmatrix}$$

double bed, the non smoking option is no longer available. The system detects increasing specificity in feature values due to information no explicitly conveyed by the user. Such an event can be used to trigger a confirmation question that enumerates the remaining possibilities. In this example, a question along the lines `We don't have any non-smoking single bed rooms. Is that okay with you?` would be generated.

4.3 Generation of Questions

The transformation is done by traversing the feature structure in depth-first order and by mapping each feature and each type encountered to a string. This generates a description for each option. The descriptions for the options are then filled in a template of the form

$$\texttt{Do you want}\langle\texttt{desc}_1\rangle, \ldots, \langle\texttt{desc}_{\texttt{n-1}}\rangle \texttt{ or } \langle\texttt{desc}_\texttt{n}\rangle?$$

The question to be generated based on the information shown in figure 6 2 would be `Do you want an Italian, a Mexican or a Greek restaurant?`. If there are more than four options, the lower bound of all types is mapped to a string. If the feature structure shown in figure 2 were to be disambiguated by this way, the question would be:

`What nationality do you want ?`

the lower bound of *italian*, *greek* and *mexican* being *nationality*. Also, when one option is preferred over the other options, a question to confirm this option is generated. An example is

`Do you want a Mexican restaurant?`

The dialogue manager determines which form of question to ask in function of the degree of ambiguity in the representation. The information provided by the user is then used to disambiguate the representations. Furthermore, the information provided by the user does not necessarily have to be one of the options mentioned in the question. Any information that serves to disambiguate the underspecified structure can be made use of. In the last example, both `no` and `no a Greek one` would be answers that serve to disambiguate (not necessarily entirely) the underspecified representation.

5 Conclusion

In this paper, we described how a domain model represented in a type hierarchy and underspecified representations of ambiguous requests can be exploited to generate clarification dialogues. The clarification questions seek complementary information from the user to disambiguate the representations. The way in which the questions are generated is data-driven. Moreover, the communicative goals are specified in terms of informational lower bounds on the representations. The described approach does not rely on a model of the dialogue itself, but on a model of the domain. This makes the human-computer interaction more flexible.

The algorithms determining form and content of the questions are decoupled from the question generation algorithms itself. This makes it possible to replace the simple generation component with more sophisticated ones in the future.

Although the dialogue can be interpreted as sequence of states of the system, it is important to notice that the states are not explicitly represented and that the system does not rely on a representation of a dialogue state. Rather, the system tries to use as much information from the input as is useful in disambiguating the representations.

The algorithms have been implemented in a dialogue system using the speech recognition engine JANUS [Waibel, 1996].

Acknowledgements

This research has been supported in part by the VODIS project funded by European Union. I would like to thank Alex Waibel, Wayne Ward and Bernhard Suhm for helpful discussions, support and advice while conducting this research.

References

[Carpenter, 1992] Bob Carpenter. *The Logic of Typed Feature Structures.* Cambridge University Press, 1992.

[Denecke, 1997] Matthias Denecke. *An Information-based Approach for Guiding Human-Computer-Interaction.* Proceedings of IJCAI 97, Japan, 1997, to appear.

[Waibel, 1996] Alex Waibel. Interactive Translation of Conversational Speech. *Computer*, 29(7), July 1996.