

The Dialog Module of the Speech Recognition and Dialog System EVAR

Marion Mast, Ralf Kompe, Franz Kummert*, Heinrich Niemann, Elmar Nöth

Friedrich-Alexander-Universität Erlangen-Nürnberg,

Lehrstuhl für Informatik 5 (Mustererkennung),

Martensstr.3, 8520 Erlangen,

GERMANY,

Fax: 49-9131-303811,

E-mail: mast@informatik.uni-erlangen.de

Abstract

This article describes the dialog module of EVAR. The dialog is seen as a sequence of dialog acts uttered by the system and the user. From a corpus of real human-human dialogs a model was extracted. The model covers all sequences of dialog acts observed in the corpus. Each dialog act is modelled as a set of pragmatic, semantic and syntactic concepts. The properties of the concepts and the current dialog state are used to identify the actual dialog act.

Each user utterance is interpreted by the system and the system reacts with the appropriate dialog act. After the user has defined his request completely, the system starts a database query and finally generates a synthesized natural language answer. The answer generation is realised with sentence masks, where the information from the database enquiry is filled in at the appropriate slots.

A dialog memory which allows the interpretation of elliptical sentences and proforms is updated after each utterance.

1 INTRODUCTION

The speech recognition and dialog system EVAR (the acronym stands for the German words for to recognize - to understand - to answer - to ask-back) is an automatic travel information system (in the domain of the German InterCity train system).

Input to the current system is continuous speech. Output of the speech recognition component is a set of up to 100 word hypotheses (depending on the duration of the utterance). The generation of word hypotheses is based on Hidden Markov Models (see [3]). The lexicon of the system contains 1081 words.

All levels of linguistic knowledge are used both to control the analysis process and for the interpretation of an utterance. All the knowledge is integrated in a homogenous knowledge base (the semantic network shell ERNEST, see [5]). The control algorithm used for the analysis is defined within ERNEST and does basically not depend on the application. It is based on the

*Franz Kummert is now with AG Angewandte Informatik, Technische Fakultät, Universität Bielefeld, 4800 Bielefeld, Germany

*A**—Algorithm. For a more detailed description of the EVAR system see [4]. Similar systems were presented e.g. in [6, 7, 8]

2 KNOWLEDGE REPRESENTATION

For the representation of knowledge the semantic network system ERNEST is used [5]. All knowledge needed for the speech understanding process and for the dialog is embedded within a single semantic network using the same representation language. Thus it is easy to propagate restrictions from all levels to support the recognition process. Nevertheless the knowledge base is easy to extend and modify because of its modularisation into *levels of abstraction*. Within a level of abstraction concepts are connected using *part_of* links. The knowledge base consists of the following levels (see Figure 1):

- The *hypothesis* level builds up the interface between speech recognition and linguistic analysis. Word hypotheses restricted to the linguistic and task specific expectations are requested from the acoustic-phonetic front-end.
- On the *syntactic* level syntactic constituents and special dialog constructions are represented but not the order of the constituents on the sentence level. In German it is characteristic for spontaneous speech that the order can be rather free.
- On the *semantic* level verb and noun frames with their deep cases according to Fillmore's deep case theory are modelled [1].
- The *pragmatic* level represents task specific knowledge.
- On the *dialog* level the dialog model as well as the dialog acts which build it up are represented.

The syntactic, semantic, and pragmatic levels especially contain a lot of knowledge to parse and interpret complex constituents specifying the time of departure and arrival.

In addition to the provision of the knowledge representation scheme ERNEST provides mechanisms to use this knowledge for the analysis process. A problem independent procedural semantics of the network language allows a flexible control of the analysis process. The *A**—Algorithm in combination with problem dependent judgement vectors is the basis of this control.

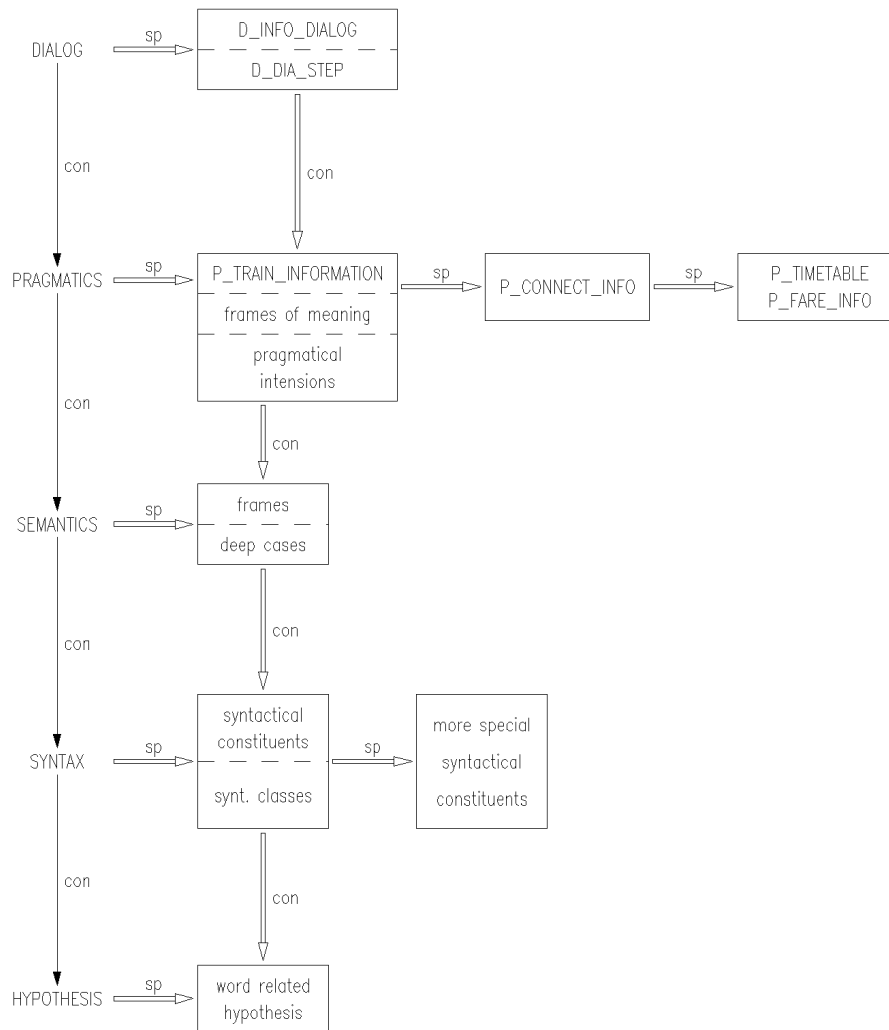


Figure 1: Network Overview (con: concrete link, sp: specialisation)

3 THE DIALOG MODEL

A user utterance has to be interpreted syntactically, semantically and pragmatically as well as in the dialog context. That comprises both the knowledge of how to behave in the situation of an information request, what kind of utterance may follow each one, but also the consideration of the dialog history in order to be able to resolve references and to find the expected answer, e.g., not to repeat information just given but to provide for a new one (see section 4).

In a user-friendly system the user should have the possibility to talk to the system without too many restrictions, i.e., almost like talking to an information officer at the station. So the dialog model must represent all dialog acts which are typical in this special situation. On the other hand, we achieved a simplification compared to real natural dialogs by guiding the user with special system utterances (see Figure 2). In Figure 2 one edge corresponds to one dialog act. The prefixes S-, U- indicate that the dialog step corresponds to a system or user utterance, respectively.

The dialog is seen as a sequence of dialog acts uttered by the system and the user. From a corpus of real human-human dialogs [2] a model was extracted containing all sequences of dialog acts observed in the corpus. Each dialog act is modelled by a set of pragmatic, semantic and syntactic concepts. The

properties of the concepts and the current dialog state are used to identify the actual dialog act. Further markers for the recognition of a dialog act are the sentence type, the intonation and metacommunicative markers.

After the greeting the user requests for information. If the information necessary for giving an answer is not given in the user's request the system starts a clarification dialog (see Figure 2). The user utterances have to be syntactically and semantically complete or they have to be incomplete in a way such that they can be completed by taking parts of prior utterances (see section 3).

The following dialog shows examples for the different dialog phases (the abbreviations of Figure 2 are used):

Greeting:

S: (S_GREETING) Hello. This is the Automatic Travel Information System EVAR.

Request:

U: (U_REQ_INFO) I want to go to Hamburg.

Request for Details:

S: (S_REQ_DET) When do you want to go to Hamburg?

U: (U_ADDITION) Tomorrow.

Request for Specification:

S: (S_REQ_SPEZ) When do you want to start tomorrow?

U: (U_ADDITION) In the afternoon.

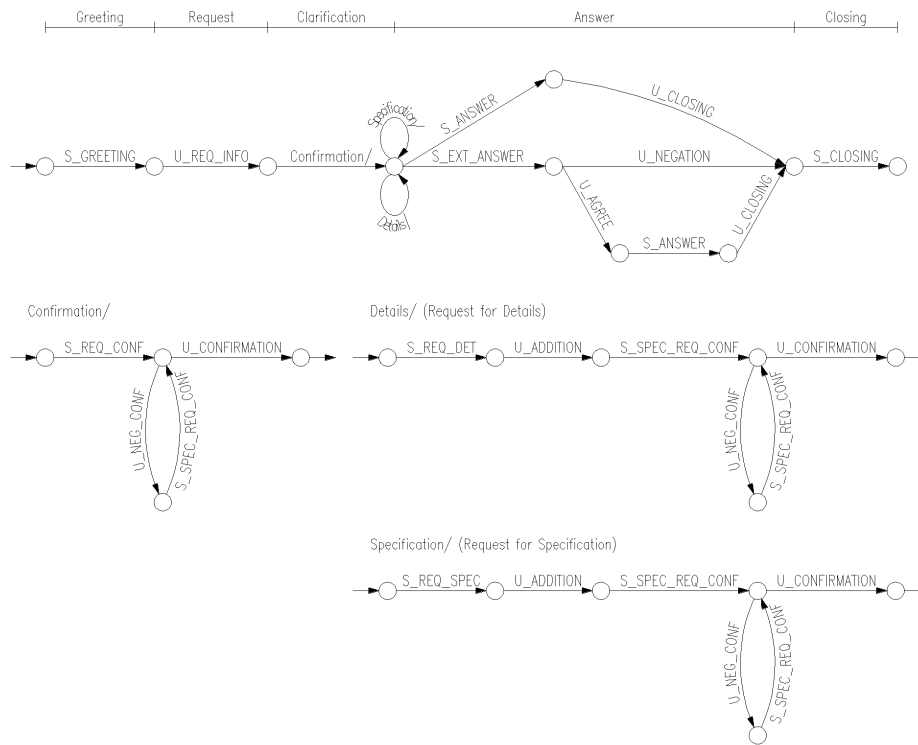


Figure 2: The Dialog Model (/ denotes a subnet)

Confirmation:

S: (S_REQ_CONF) You want to go tomorrow in the afternoon from Nürnberg to Hamburg?¹

U: (U_CONFIRMATION) Yes, to Hamburg.

Answer:

S: (S_EXT_ANSWER) You can take the train at 14h15. Do you want a later train?

U: (U_NEGATION) No thanks.

Closing:

S: (S_CLOSING) Thanks for calling the Automatic Travel Information System, good bye.

4 DIALOG HISTORY

A special problem within a dialog situation where partners presuppose a certain amount of common contextual and situational knowledge is the determination of possible referential objects in the real world. This is done with the help of a memory of the dialog history. The resolution of anaphorically used constituents, i.e. constituents referring back to some previously mentioned objects is of special interest. In dialogs in the given domain two types of anaphora often occur:

After an utterance like: When is there a train to Hamburg?

1. When does *it* arrive in Hamburg? (pronoun)
2. When does the train arrive *there*? (pronominal adverb)

Another important feature especially for speech is the frequent usage of elliptical constructions. Currently we concentrate on the analysis of ellipses which are generated using the linguistic constructions of the prior utterance. Two types are distinguished:

¹Nürnberg is the default place of departure.

1. The “syntactic” ellipses, i.e., grammatically incomplete simple constituents where the head has to be taken from outside of the linguistic context, for example
Is it *the last (one)*? (nominal ellipsis; in German the ‘one’ is not used).
2. The “semantic” ellipses, i.e. grammatically incomplete sentences where parts of the sentence like the verb or some of its actants are taken out of the linguistic context, for example
S: You can take the intercity train at 13h30.
U: *13h30*.

Both types can cooccur within a single sentence.

For the resolution of all references the whole dialog must be available. The dialog history consists of three components:

1. For anaphora all relevant objects during the dialog are stored in the dialog memory. If a proform occurs in an utterance, which refers to an object, this list is searched for a fitting object.
2. For the resolution of ellipses the actual dialog state has to be stored in the dialog history that means all constituents which could be missing in the following utterance are available during the analysis of the following utterance.
3. In the third component all information given to the current database request is available until the request is answered.

5 DIALOG CONTROL

As mentioned before, the dialog model is represented by a concept in the semantic network. Parts of this concept are all dialog acts which build it up. Each dialog act itself is realized as a concept in the semantic network. Its properties are specified by attributes. The order in which they can appear is

represented by an adjacency matrix. But only after a system utterance every user dialog act which is possible according to this order can follow. If the adjacency matrix allows more than one system dialog act, then one of these dialog acts is selected dynamically based on the actual dialog history. For example after a user request for information by the user either a request for specification (if one obligatory parameter is missing), a request for details (if a parameter has to be stated more precise), or a request for confirmation can follow (see figure 2).

If the system doesn't succeed in interpreting the actual user utterance after a certain analysis time, the system asks for repetition of the last utterance. With this mechanism the dialog can handle bad recognition results.

The dialog control can realize different *dialog strategies*. For example in a system trained for one speaker or in a NL-mode, requests for confirmations are not that necessary. Whereas in a multi-speaker system or a speaker-independent system, it is better to ask for confirmation.

6 DATABASE ACCESS

To enable the system to answer requests in the domain of train time tables and prices, database access is needed. Input to the database query are the parameters the user gives about the connection he needs. These are at least the destination and an interval for the departure or arrival time. This time interval was derived from a possibly abstract and complex user specification using the semantic and pragmatic knowledge. For the departure place the system uses a default (the city in which the system is located), if nothing else was uttered. If one obligatory parameter is missing, the dialog module has to start a request for it.

Before the database retrieval, several consistency checks are performed, e.g., the given time interval should not exceed a certain limit, otherwise the set of retrieved connections will be too large. Then the database is searched for all suitable connections which match the given parameters.

The system uses the same database as the German railway company, i.e., the HaFas-Database².

7 ANSWER GENERATION

The emphasis in the developed system is on the analysis of utterances in task-oriented dialogs in the domain of information provision services. To enable the system to communicate in a spoken dialog with the user, and not only to answer single questions like in a question-answer system not only a dialog component but also a component is needed, which generates answers and system questions.

For the answer generation sentence masks are used for each dialog act. Besides some metacommunicative acts, which control the phatic communication, dialog acts which are concerned with the domain are needed. The answer schemes for the latter acts need to be updated during the dialog.

Example: Request for confirmation of destination and time of arrival after the request for information of the user:

*"Ich möchte morgen nachmittag in München ankommen."
("I want to arrive in Munich tomorrow in the afternoon.")*

²The database was developed by HaCon, Hannoversche Consulting für Verkehrswesen, Transporttechnik und Elektronische Datenverarbeitung GmbH, Hannover, Germany.

In the scheme for requests for confirmation

*"Sie wollen in ORT ZEIT ankommen?"
("You want to arrive in PLACE TIME?")*

the variables for destination ORT (PLACE) and arrival time ZEIT (TIME) have to be replaced by the actual parameters to produce the following output:

*"Sie wollen in München morgen nachmittag Uhr ankommen?"
("You want to arrive in Munich tomorrow in the afternoon?")*

The time uttered by the user is repeated by the system rather than transforming it into an absolute time. Apart from times and places the result of the database request, e.g., a connection, has to be filled in an answer scheme. The complete text of an answer is sent to a text-to-speech system (developed by the Daimler Benz AG). No prerecorded signals are used.

8 EXPERIMENTS

For preliminary experiments a more restricted dialog model was used which consists of up to 5 dialog acts. After the initial user request for confirmation which can be preceded by a greeting, the system asks for confirmation or for parameters which are needed for the database enquiry. Then the system starts a database request, fills an answer pattern with the given information and generates an answer with the speech synthesizer.

A speaker-dependent version of the acoustic front-end was used which was trained on 100 domain specific and 200 phonetically balanced sentences. A bigram model of perplexity 111 was used. The generation of the word hypotheses after recording each utterance was 3.53 times realtime. The word accuracy was 90.9% (91.1% of the words were correct and 54.2% of the sentences were correct).

85 dialogs with a total of 170 user utterances were tested. 68 of the dialogs were completed successfully that is the system provided the correct train connection. In 3 of these cases a successful completion was possible after the system failed to analyze the user request but the user corrected the system after the request for confirmation (see confirmation subnet in figure 2). 17 dialogs were not completed successfully due to memory limitations or an incorrect analysis. The average time to complete a dialog was 3:57 minutes. The average CPU time for the linguistic analysis to complete a dialog was 1:32 minutes.

Acknowledgements

This work was supported by the German Research Foundation (DFG). Only the authors are responsible for the contents of this publication.

References

- [1] Fillmore, C.: *A Case for Case*. In Bach, E.; Harms, R. T., editors: *Universals in Linguistic Theory*, pages 1-88. Holt, Rinehart and Winston, New York, 1968.
- [2] Hitzenberger, L.; Ulbrand, R.; Kritzenberger, H.; Wenzel, P.: *FACID - Fachsprachlicher Corpus informationsabfragender Dialoge. Endbericht*. Universität Regensburg, Regensburg, 1986.
- [3] Kuhn, T.; Niemann, H.; Schukat-Talamazzini, E.; Eckert, W.; Rieck, S.: *Context-dependent modeling in a two-stage HMM word recognizer for continuous speech*. In Vandewalle, J.; Oosterlinck, A., editors: *EUSIPCO'92-Proceedings of the EUSIPCO 92 Conference*. Elsevier science publisher B.V, 1992.

- [4] Niemann, H.; Brietzmann, A.; Ehrlich, U.; Posch, S.; Regel, P.; Sagerer, G.; Salzbrunn, R.; Schukat-Talamazzini, G.: *A Knowledge Based Speech Understanding System*. *Int. J. Pattern Recognition and Artificial Intelligence*, 2(2):321–350, 1988.
- [5] Niemann, H.; Sagerer, G.; Schröder, S.; Kummert, F.: *ERNEST: A Semantic Network System for Pattern Understanding*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:883–905, 1990.
- [6] Pieraccini92, R.; Tzoukermann, E.; Gorelov, Z.; Levin, E.; Lee, C.; Gauvin, J.: *Progress Report on the Chronus System: ATIS Benchmark Results*. In: *Fifth DARPA Workshop on Speech and Natural Language, 23 - 26. Februar 92*, Arden Conference Center, Harriman, NY. 1992.
- [7] Young, S. J.; Proctor, C. E.: *The Design and Implementation of Dialogue Control in Voice Operated Database Inquiry Systems*. *Computer Speech & Language*, 3(4):329–353, 1989.
- [8] Young, S. R.; Hauptmann, A. G.; Ward, W. H.; Smith, E. T.; Werner, P.: *High Level Knowledge Sources in Usable Speech Recognition Systems*. *Communications of the ACM*, 32:183–194, 1989.