From Proc. International Symposium on Spoken Dialogue (ISSD 98), Sydney, 1998, pp. 19–26

The Erlangen Spoken Dialogue System EVAR: A State–of–the–Art Information Retrieval System

F. Gallwitz[1], *M. Aretoulaki*[2], *M. Boros*[2], *J. Haas*[1], *S. Harbeck*[1], *R. Huber*[1], *H. Niemann*[1], *E. Nöth*[1]

[1]University of Erlangen–Nuremberg, Dept. of Pattern Recognition
Martensstr. 3, D-91058 Erlangen,

[2]Bavarian Research Centre for Knowledge–Based Systems (FORWISS)
Am Weichselgarten 7, D-91058 Erlangen, Germany

gallwitz@informatik.uni-erlangen.de, aretoula@forwiss.de

ABSTRACT

In this paper, we present an overview of the spoken dialogue system EVAR that was developed at the University of Erlangen. In January 1994, it became accessible over telephone line and could answer inquiries in the German language about German InterCity train connections. It has since been continuously improved and extended, including some unique features, such as the processing of out–of–vocabulary words and a flexible dialogue strategy that adapts to the quality of the recognition of the user input. In fact, several different versions of the system have emerged, i.e. a subway information system, train and flight information systems in different languages, and an integrated multilingual and multifunctional system which covers German and 3 additional languages in parallel. Current research focuses on the introduction of stochastic models into the semantic analysis, on the direct integration of prosodic information into the word recognition process, on the detection of user emotion, and on multilinguality and multifunctionality.

1. Introduction

The spoken dialogue system EVAR was developed at the University of Erlangen over a period of almost 20 years. Different system architectures have been implemented and evaluated, and intensive research has been performed in the areas of word recognition, linguistic analysis, knowledge representation, dialogue management, and prosodic analysis. To our knowledge, EVAR was the first spoken dialogue system in the German language that was made available to the general public when it was connected to the telephone line in January 1994 (EVAR's phone number: +49 9131 16287). Since that time, a corpus of approximately 3000 spontaneous human–machine dialogues has been compiled, most of them in the train timetable information domain, involving mainly naive users who are not familiar with the speech recognition and understanding technology. These dialogues are constantly transcribed and used for retraining, improving, and evaluating the various system components.

EVAR ("*Erkennen, Verstehen, Antworten, Rückfragen*", or "Recognize, Understand, Answer, Ask back") was designed as a research platform, where current developments can be implemented and evaluated with real users, and speech data can be collected. The dialogue strategy adopted is less strict than in the case of systems that are intended for commercial applications, because to us, the collection of actual spontaneous speech data is more important than an optimal dialogue success rate. Nevertheless, word accuracy, semantic accuracy, and dialogue success rates have all considerably increased over the last few years, partly due to improved algorithms and dialogue strategies, and partly to the increasing availability of training data.

An overview of the system architecture is given in Section 2, where a brief description is also provided of the word recognizer, the linguistic processor, the dialogue manager, the speech synthesis module, and the WWW database interface. A number of the unique features of EVAR is then outlined. In Section 3, we explain how the EVAR dialogue strategy adapts to the acoustic channel and the cooperativeness of the user. In Section 4, we explain how out–of–vocabulary words are detected and classified by the word recognizer and how this information is used to generate an appropriate system reaction. In Section 5, we briefly report on the evolution of EVAR into a multilingual and multifunctional system that automatically detects the appropriate language and domain.

The subsequent sections focus on some recent developments that will soon be integrated into EVAR. Stochastic methods for semantic analysis are discussed in Section 6, which should complement and enhance the traditional linguistic methods. A new approach involving the integrated recognition of words and prosodic boundaries is presented in Section 7, which allows the implicit detection of syntactic structure during the word recognition process. Finally, we outline our current research on the detection of user emotion, which is an essential issue to be dealt with if

This research was funded by the DFG (German Research Foundation) under contract number 810 939-9, by the EC in the framework of the Copernicus project COP-1634 (SQEL), and by the German Federal Ministry of Education, Science, Research and Technology (*BMBF*) in the framework of the VERBMOBIL Project under Grant 01 IV 701 K5. The responsibility for the contents of this article lies with the authors.

spoken dialogue systems are to be used in real–world applications.

2. System Architecture

The architecture of EVAR is depicted in Figure 1. User utterances are first digitalized by an AD/DA converter. Then word recognition is performed and the best word chain (e.g. "*I would like to go to Frankfurt*"), or alternatively a word graph, is handed on to the linguistic processor. The linguistic processor extracts a set of semantic concepts (semantic attribute–value pairs) from the word recognizer result (e.g. `[goalcity:frankfurt]`) and forwards them to the dialogue manager. The dialogue manager checks whether all necessary parameters are available and, if so, sends a query to the application database. Depending on the dialogue history and the current dialogue strategy, the user is asked to confirm the parameter (e.g. "*You want to go to Frankfurt?*") and/or another parameter is requested (e.g. "*What time would you like to leave?*"); otherwise the result of the database search is verbalized. The generated message is then synthesized by a text–to–speech module and played to the user over the AD/DA converter that is connected to the telephone line.

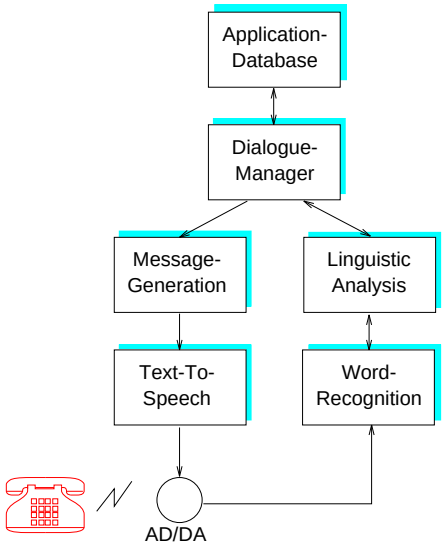

Figure 1: The basic architecture of EVAR

2.1. The Word Recognizer

EVAR uses a two–pass word recognition module that is based on semi–continuous Hidden Markov Models (SCHMM) and n-gram language models. The basic architecture of the recognizer is depicted in Figure 2; more details on the recognizer can be found in [9].

The HMM word models consist of *polyphone* subword units, which are a generalisation of the widely–used triphone models and were first introduced in [19]. The size of the phonetic context covered by polyphone models depends on the amount of training data available for each phone context, with common words being automatically

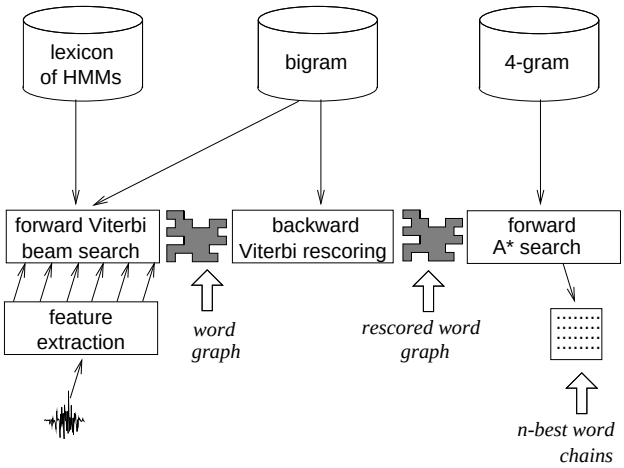

Figure 2: Architecture of the EVAR word recognizer

represented as whole–word models. Currently, we use an improved version of the polyphone subword unit called the *generalised polyphone*, which combines the basic concept of structuring phones with different context sizes in a generalisation tree with a hierarchy of phone superclasses. This leads to a further significant reduction of word error rates.

The recognizer uses a set of stochastic language models that are activated by the prediction forwarded by the dialogue manager. For example, if the dialogue manager expects the departure city, this increases the probability of a user utterance, such as "*I would like to leave from Hamburg*" or simply "*Hamburg*". Nevertheless, the stochastic language model will also accept utterances, such as "*No, at ten o'clock*". This can also be an adequate response by the user, because EVAR may ask for the confirmation of one or more parameters along with the value of a new parameter in a single dialogue turn, e.g. "*You want to leave around twelve o'clock. Where would you like to leave from?*". The dialogue–step–dependent stochastic language models are trained automatically using subsets of all recorded user utterances, which have been labelled with the corresponding dialogue manager prediction.

2.2. The Linguistic Processor

The linguistic processing component (LP) of EVAR comprises the parser and a domain–dependent linguistic knowledge base on which the parser operates. Accepting the output of the speech recognizer as its input, its task is to build up a semantic representation of the user's utterance and forward this representation to the dialogue module for interpretation. The semantic representation formalism further employed is the *S*emantic *I*nterface *L*anguage (SIL)[15].

The agenda–driven chart parser used achieves robustness by means of the island parsing technique. This renders it able to select partial results out of the chart, when no global parsing result has been computed [16]. In addition, the parser can operate either on graphs of word hypotheses or on the best recognized word string, depending on

the word recognizer employed. This means that in the case of a word graph, the parser has to look for the best–score, grammatically correct path in the graph and build up its semantic representation, whereas in the case of a word string, syntactic and semantic analysis can be directly carried out. Either way, analysis is based on the syntactic and semantic knowledge stored in the linguistic knowledge base.

Linguistic knowledge is defined in terms of Unification Categorial Grammar (UCG) [24]. This formalism combines unification with categorial approaches to grammar. Being a categorial grammar, the number of rules employed is restricted to a few basic rules of combination, while most of the combinatorics of words is encoded in the lexical categories themselves. Lexical entries are represented as complex feature structures, which are merged by means of simple unification.

2.3. Speech Synthesis

The text–to–speech module currently used in EVAR involves the simple concatenation of prerecorded phrases and words. This leads to good intelligibility and — at least for the longer fragments — to a natural prosody, but the effort required in changing the domain or the dialogue strategy is large. The prerecorded speech signals are slightly warped in order to make them sound machine–like, because we want to assure callers that they are indeed talking to a machine.

The quality of the speech synthesis has proven to have considerable impact on the speaking style of the users. When we used an even simpler concatenation method of prerecorded words, with noticeable silence periods at the word boundaries, many users imitated this speaking style, either deliberately or unconsciously. This means that a speech corpus compiled using a spoken dialogue system will always reflect the speech synthesis algorithm chosen. Thus, a modification of this module can result in a decline in word recognition accuracy. The same is also true for changes in the wording of system utterances, as they also have an impact on the actual formulation of user utterances.

2.4. The Dialogue Manager

The role of the Dialogue Manager (DMan) in EVAR is: (a) to interpret the semantic representation of the user utterance that was forwarded by the linguistic processor, in the light of the application, the domain, and the specific dialogue history and (b) to plan the content and the formulation of an appropriate response. These tasks are carried out by various independent but closely collaborating submodules (Figure 3).

The Linguistic Interface (LI) interacts directly with the Parser and isolates the information that is relevant to the task: the parameters necessary for database access (e.g. place and time of departure and arrival) and various dialogue markers which determine the state of the dialogue and influence its progression (e.g. 'Yes', 'No', 'Thanks, bye'). The Belief Module (BM) is where the 'anchoring', or disambiguation, of underspecified semantic representations takes place based on the predictions available regarding the development of the dialogue. This is where an isolated city name (e.g. 'Munich') in an elliptical user utterance is interpreted as either the goal or the source location in the user requirements profile. This is also where anaphors and relative expressions of place and time are clarified. The most central component of DMan is the Dialogue Module (DM). This keeps track of the progression of the dialogue in terms of system and user dialogue acts, their translation into system goals and the degree of satisfaction of the latter. An Augmented Transition Network (ATN) description of the possible transitions between dialogue states is used to generate expectations about the continuation of the dialogue, in terms of both user and system acts (Section 3). The Task Module (TM), where the interface to the domain database is maintained, holds information about the identity and the number of the task parameters that should be specified by the user before the database can be accessed: goalcity, sourcecity, date, and goaltime or sourcetime, in the case of train enquiries. Finally, the Message Planner (MP) determines the next system utterance on the basis of the report provided by DM on the state of the dialogue. Accordingly, it may either generate a request for information or confirmation, when the values for certain of the parameters are still missing or contended, respectively; or it may present to the user the database entries matching the query, when sufficient specifications have been supplied to initiate a search.

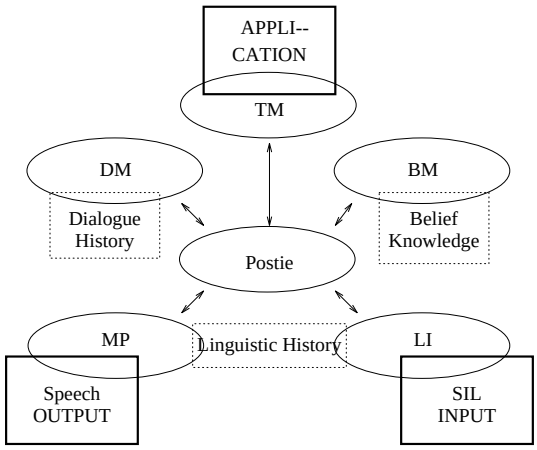

Figure 3: The EVAR Dialogue Manager Processing Flow

2.5. WWW Database Access

The long–term vision regarding EVAR is its evolution into a multimodal environment, where text, speech, and even image processing are integrated over the phone and the internet. A step in this direction has been the development of a search engine that poses the user's query to multiple travel information databases on the WWW [2]. The search engine constitutes the interface between the dialogue system and the WWW databases. To date, the following databases are accessed: German Railways (*DB*),

Lufthansa, and Swiss Railways (*SBB*). During the search, a number of dynamic HTML documents are created and accessed holding the intermediate results collected. These are temporarily saved in a local cache which is continually updated until the search is ended. The engine also provides the facility for a number of local databases to be set–up for regularly–accessed data. Although the retrieved entries are filtered according to the parameters specified by the user in the course of the dialogue, constraints can be relaxed, when the initial query does not match any of the stored data.

3. Flexible and Adaptive Dialogue Strategy

One of the distinctive traits of EVAR is the adoption therein of 'open' dialogue strategies, allowing the user to freely formulate their queries and carry out the transaction quite flexibly. The user is allowed to take the initiative regarding the order in which task parameter specification takes place and is also usually able to change the current subgoal of the interaction; e.g. in correcting a parameter that has already been dealt with, at a time when the system is expecting information about another parameter. This contrasts to the more common approach of presenting the user with menus to which they have to comply and answer with yes or no. In this sense, EVAR is quite intelligent. As a result, however, there are more possibilities regarding the content of the next user utterance, thus increasing the probability of misrecognitions and misunderstandings. To remedy this, the system tries to always confirm each task parameter as they are specified by the user. This defensive strategy safeguards that the correct database entry will be retrieved at the expense of interaction speed.

Confirmation can be both explicit and implicit, depending on whether or not there has been a history of failures in the current dialogue (Figure 4). A potential correction by the user initiates a clarification subdialogue, which can contribute to the repair of the most crucial errors and, hence, to the successful completion of the task in most cases. This is effected by asking the user to repeat or confirm a parameter on its own, or to spell it in the worst case, thereby restricting the user's usual freedom. Normally, however, the user is expected to confirm a parameter indirectly in conjunction with the specification of other parameters, which contributes to greater user satisfaction [5, 2]. Consequently, EVAR can dynamically modify its communicative and repair strategies employing the user's reactions in the course of the dialogue as a guide and the frequency of conflict between the system's beliefs and the user's goals.

4. Out–of–Vocabulary Word Processing

One of the most important causes of failure in spoken dialogue systems is usually neglected: the problem of words that are not covered by the system's vocabulary (Out–Of–Vocabulary or OOV words). In such a case, the word

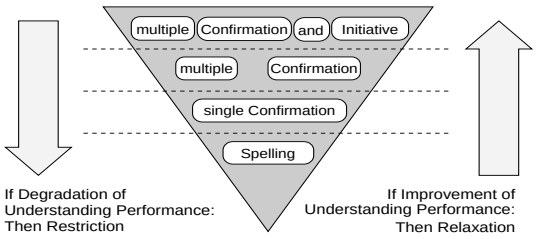

Figure 4: Flexible Dialogue Strategies in EVAR

recognizer usually recognizes one or more different words with a similar acoustic profile to the unknown. These misrecognitions often result in possibly irreparable misunderstandings between the user and the system. This is due to the fact that users rarely realise that they have crossed the boundaries of the system's knowledge, but just notice its suddenly weird behaviour. Therefore, it is desireable to have the system detect unknown words and inform the user about them so that they might correct the error. This will increase not only the dialogue success rates but also the acceptability of the system to the user.

4.1. Detection and Classification of OOV Words

In [8], we presented an approach to the detection of OOV words which implicitly provides information on the word category. This involves the integration of both detection *and* classification of OOV words directly into the recognition process of an HMM–based word recognizer. With our approach, acoustic as well as language model information can be used for the purpose of classifying OOV words into different word categories. Currently the same acoustic models are used for all OOV words; only language model information contributes to the assignment of a category to each.

The basic idea behind our approach is to build language models for the recognition of OOV words that are based on a system of word categories. Emission probabilities of OOV words are then estimated for each word category. Even if we include in our vocabulary all words of a category that were observed in the training sample, there is still a certain probability of observing new words of the same category in an independent test sample or in future utterances. This probability can be estimated from the training sample itself. Details on the calculation of the OOV emission probabilities were given in [8].

For most of our linguistically–motivated word categories, the OOV probability is 0, because they describe a finite set of words. In the timetable inquiry domain, there are 5 word categories that are practically infinite (e.g. CITY, REGION, SURNAME). In addition, a category has been defined for rare words that do not fall under any other category (OOV probability 73%) and another one for garbage (e.g. word fragments, OOV probability 100%).

After integrating OOV probabilities into the language model, the latter has to be combined with one or several

acoustic models for OOV words. Simple 'flat' acoustic models can be used for this purpose, as well as more enhanced models based on phone- or syllable–grammars. In EVAR, we use 'flat' acoustic models that are constructed by averaging the acoustic distributions of all phone models in the training corpus.

For the experiments described in [3], an encouraging increase in both word accuracy and semantic concept accuracy was achieved. The corresponding error rates dropped by 5% and 7%, respectively. The OOV words that were detected by the word recognizer were correctly classified in 94% of the cases.

4.2. Extensions to the Linguistic Processor

Word strings delivered by the OOV–extended recognizer will contain the corresponding information if an OOV word has been detected. In order to make this information accessible to the dialogue manager, the LP has to include it into the semantic representation that is passed on to the dialogue manager. The system will then be capable of reacting appropriately to an OOV word. Details on the linguistic processing of OOV words in EVAR can be found in [3].

4.3. Extensions to the Dialogue Manager

The extension of the word recognizer and the LP of EVAR with meta–knowledge about the occurrence of OOV words has led to the design of new dialogue strategies that take this extra information into account and are adopted on–line in the presence such a word. Thus, two new dialogue states were incorporated in the corresponding ATN description of the dialogue structure, which accommodate alternative state transitions in the DMan accordingly: (a) **REPEAT_PARAM** is used to ascertain that an OOV word was indeed uttered, in order to avoid false alarms. It provides a first warning to the user that there may be a problem and asks them to repeat just the parameter value involved. (b) **WARN** follows the default repair mode **SPELL** and involves the notification of the user about the cause of failure so that they can either hang up (**CLOSE**) or pose a different query (**FURTHER_INFO**).

This is illustrated in the following example dialogue:

```
User:   I want to go to <Brussels> oov_city.
          [goalcity:oov_city]
System: I think the information you require is not
        covered by our database. Could you, please,
        repeat the name of the city you want to go
        to?
          [system goal: REPEAT_PARAM]
User:   <Brussels> oov_city.
System: Could you please spell the name of this
        city?
          [system goal: SPELL]
User:   <B-r-u-s-s-e-l-s> oov_city.
System: Unfortunately, there is no information on
        train connections for the city you want. Our
        database only covers German cities. Would
        you like to proceed with a different query?
          [system goal: WARN]
User:   Yes.
System: What exactly would you like to know?
          [system goal: FURTHER_INFO]
```

4.4. Evaluation of the OOV–extended EVAR

'Transaction Success' (TS) is an evaluation metric used in the context of dialogue systems which refers to whether or not the user got the information they were asking for in an effortless manner. 'Dialogue Completion' (DC) is another metric relating to the smoothness of closing of the dialogue, respective of whether or not the task was carried out successfully. A study of the protocols for the majority of the dialogues collected with EVAR since it was extended with OOV information (September 1997 – May 1998: 282 [sub]dialogues) [2] showed that TS had increased from *47%* to *59%*. Moreover, a DC rate of *69%* was established for the new system version (the corresponding rate for the old version was not available). It has also emerged that the correct treatment of OOVs by EVAR is associated mainly with completed and successful dialogues, *88%* and *93%* respectively, thus justifying the incorporation of this new functionality in the system.

5. Multilinguality and Multifunctionality

Within the EC Copernicus–project SQEL, EVAR was extended with respect to multilinguality and multifunctionality [1]. The current SQEL–demonstrator can handle four different languages and domains: Czech (Czech train connections), German (German train connections), Slovak (Slovak train connections), and Slovenian (European flights). The system starts up in German with a German opening phrase, but the user is free to use any of the implemented languages. The language is identified implicitly by a multilingual word recognizer [1]. During the recognizer beam search, search paths related to other languages are normally eliminated in the first few seconds, thus keeping the overhead low compared to running four monolingual recognizers in parallel (Figure 5).

After completing the recognition process, the recognizer passes to the linguistic processor both the identity of the language spoken and the best matching word chain. The recognition results for the multilingual recognizer are almost as good as when the monolingual version is run for the language that was spoken. Once the language has been identified by the word recognizer, it is associated with the corresponding domain, which calls the appropriate database and task parameters. All domains are already accommodated for in the generic concept ontology used in EVAR. This ontology covers concepts such as source and goal location, departure and arrival time and date of travel. The existence of such language–independent semantic units has meant that porting the system to a new

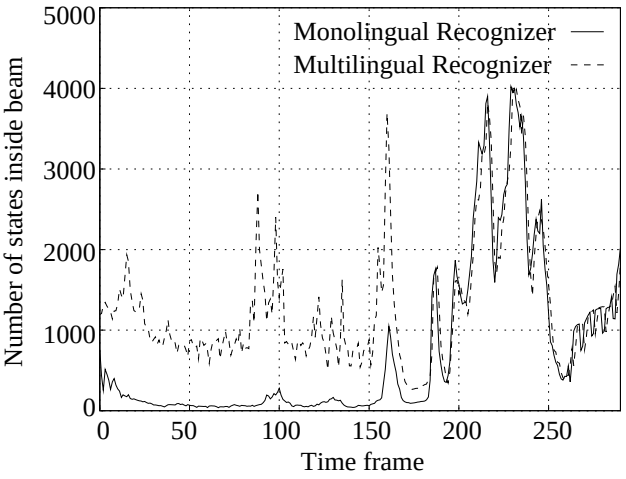

Figure 5: Implicit language identification by the word recognizer. In this example, the language spoken is identified after approx. 1.8 seconds (180 time frames). At this point, the multilingual recognizer starts to function as a monolingual recognizer.

language involves mainly the development of new lexica and grammars for the analysis and the generation phases (apart from the word recognizers) and not an extensive restructuring of the interpretation process within the dialogue manager. This is because the dialogue manager of EVAR is sufficiently flexible to switch between the different domains, i.e. to the appropriate parser, generator, and application database for each language and domain.

6. Stochastic Methods for Semantic Analysis

The motivation behind the use of statistical methods in speech processing tasks has been the improved and sometimes more efficient analysis of the data in hand, while also reducing the amount of expert knowledge that needs to be incorporated in the corresponding system. The result is the fast and straightforward adaptation to new tasks and domains. This is why one of the major research areas we currently focus on is the introduction of statistical methods at the level of semantic and linguistic analysis. Two different methods are described here that are to be used in EVAR during linguistic analysis.

The basic assumption behind using stochastic methods in semantic analysis is that the analysis effort can be greatly reduced when operating in a restricted domain such as train timetable information. It is not necessary to carry out a complete linguistic analysis, because it is sufficent for the system to locate and understand those parts of an utterance which exercise influence on the database query to be performed, as well as those parts influencing the dialogue structure. For example, it is essential to know the city of departure but it is completely irrelevant to know who wants to travel or why someone needs a connection. Those parts that are important to the dialogue system are called semantic attributes and semantic/pragmatic analysis could be limited to them.

In [4], a new concept was introduced of using stochastic models for semantic analysis. The idea is to avoid using a grammar that describes all important parts at once, and to write instead small partial grammars, each of them covering a special attribute; e.g. a grammar that is able to analyse time expressions. Given that not every utterance includes all semantic attributes, we use stochastic methods to predict the occurence of attributes in the current utterance, only using partial grammars for the semantic analysis of the predicted attributes. The advantage is that partial grammars are easy to maintain and reuseable in other systems for different tasks. In addition, the analysis is carried out faster without a decrease in accuracy.

As stated before, we have developed two different methods for semantic attribute prediction. The first one uses n–gram language models [18] to decide whether or not a specific attribute is included in the current utterance. For this purpose, we divide our corpus into two subsets for each attribute. The first one consists of all sentences comprising this attribute, the second one is made up of all other utterances. With these two subsets, two language models are trained for each attribute, signaling its occurrence or its absence, respectively. During the analysis phase, the probability of the recognized word sequence is calculated with respect to each language model, and, for each attribute, a decision is made according to the higher probability. Only if the language model denoting the occurence of this attribute wins, the corresponding partial grammar becomes active. The experiments described in detail in [11] show prediction rates of approximately 90% in average for three different attributes (time, date and city expressions).

Our second approach to stochastic semantic analysis is based on standard and higher order HMMs. It is the basic assumption underlying this model, that each word w_i in a word chain $\boldsymbol{w}$ can be assigned to precisely one semantic attribute. This can be described in a formal manner by defining an assignment function $\boldsymbol{\zeta}$ which takes the word chain $\boldsymbol{w}$ as input and produces for each word the corresponding attribute. This is illustrated in the following example:

I	*want*	*to*	*go*	*from*	*Munich*	*to*	*Hamburg*
NIL	NIL	NIL	NIL	SOURCE	SOURCE	GOAL	GOAL

Here, NIL marks those words that are irrelevant to the application, SOURCE is a marker for the city of departure, and GOAL for the city of arrival. Resting on this basic assumption, two tied statistical processes are used as a model, one describing the occurrence of attributes and the other for relating words to attributes. As the assignment function $\boldsymbol{\zeta}$ is unknown during training — only the set of attributes each utterance includes is annotated — we use the EM algorithm to derive estimation formulas for the discrete density functions involved. These estimation formulas are generalised versions of the well known Baum–Welch reestimation formulas. For a detailed discussion of this approach, cf. [10]. Using this method, we obtain detection

rates of 95% for time, date and city expressions.

7. Integrated Recognition of Words and Prosodic Boundaries

It is widely recognized that prosodic information is a useful source of information in speech understanding. Nevertheless, most existing systems do not make use of it. The most probable reason is, that these systems can usually only handle fairly short and simple utterances in a very restricted domain, where prosodic functions — such as the marking of *sentence mood*, *accentuation*, and *phrasing* — are believed to be of little importance. In such applications, efficient linguistic processing is possible without any prosodic information about the structure of the utterance, and the prosodic marking of the sentence mood and accentuation are mostly redundant. An acceptable system performance can also be achieved, if prosodic information is totally ignored. This is also the case in the EVAR domain.

When moving to more complex tasks, such as in VERBMOBIL human–to–human spontaneous speech translation [22], prosody becomes an important issue. In VERBMOBIL, utterances tend to be considerably longer than those in EVAR; the average number of words per utterance in EVAR dialogues is only 3, with an average of 7 words in the first utterance, whereas an average VERBMOBIL utterance is made up of 22 words. This makes it particularly important to identify prosodic phrase boundaries. In spontaneous speech, prosodic boundaries are even more crucial in understanding an utterance than punctuation marks are in written language. Words which "belong together" from the viewpoint of meaning are grouped into *prosodic phrases*, and it is widely agreed upon that there is a close correspondence between prosodic and syntactic phrase boundaries [20, 6, 23, 13].

The VERBMOBIL system uses a prosodic classifier that determines phrase boundaries based on the word recognizer output and the speech signal. In [13], the parsing of word graphs computed on VERBMOBIL spontaneous speech data was sped up by 92%, while the number of parse trees could be reduced by 96% with the use of automatically determined prosodic phrase boundaries [14, 17].

In [7], we propose the direct integration of the classification of phrase boundaries into the word recognition process. HMMs are used to model phrase boundaries, which are also integrated into the stochastic language model. The word recognizer then determines the optimal sequence of words and boundaries. In the VERBMOBIL domain, even without additional prosodic features, we obtain phrase boundary recognition rates that are comparable to those achieved with the separate prosodic classifier introduced above. At the same time, a word error rate reduction of 4% is attained without any increase in computational effort [7].

This approach to phrase boundary classification renders prosodic information available already during the word recognition process. It is assumed, that integrating prosodic boundary information into EVAR will be especially useful for recognizing utterances, such as *"No, no, on sunday at eight, not on monday at eight"*. Moreover, it can be observed in the EVAR corpus that the boundaries between semantic attributes are often marked prosodically, e.g. *"On thursday, from Hamburg, around eight o'clock"*. Thus, we are currently conducting experiments using different types of boundary labels on the EVAR corpus and are confident that similar improvements will be effected to those in the *Verbmobil* domain. We will also investigate the usefulness of prosodic boundary information to the linguistic processor.

8. User Emotion

Just as people kick soda vending machines when these don't work, it can be observed in the EVAR speech corpus that users get angry at spoken dialogue systems, when a dialogue with such a system goes wrong. In the context of call–center applications, it is important to identify such a situation and to initiate an appropriate reaction, such as referring the customer to a human operator or starting a clarification sub–dialogue, if one does not want to lose a potential customer forever. The detection of emotion and an adequate reaction to an angry user will certainly lead to a higher degree of acceptance of the system.

Even though there are many different emotions that can be expressed in human speech — such as sadness, joy, or fear — we are currently only interested in the distinction between anger and normal speaking. Other types of emotion are most probably not relevant to the implementation of the emotion detector in spoken dialogue systems, such as EVAR. Besides, it is more difficult to distinguish between emotions, such as joy and anger [21].

Emotion can be expressed, at least, in two verbal ways (in addition to non–verbal cues like body language): by the lexical information carried by certain words of an utterance, e.g. swear words; and by way of acoustic prosodic cues, such as sharp changes in the loudness and/or the fundamental frequency (F_0) of the speech signal, as well as changes of the duration values. All these changes do not have to occur in a single utterance. They can also take place in conjunction with earlier utterances of a dialogue. Furthermore, emotion can be expressed by means of a combination of the above parameters. We are currently concentrating on the use of acoustic prosodic cues to locate emotional utterances, without using the lexical information of words.

For the classification of emotion 276 prosodic features are used in our approach, based mainly on durational cues and fundamental frequency and energy contours. An artificial neural network is used as classifier. On a training set of 1530 utterances and a test set of 300 utterances, we attained a precision rate of 87% and a recall rate of 92% for the detection of angry vs. normal speaking style (cf. [12]).

9. Conclusion and Future Work

Although the first commercial spoken dialogue systems for train timetable information are beginning to emerge, running EVAR on the public telephone line still gives us the valuable opportunity of assessing newly developed techniques on real–world users. In this paper, we presented EVAR and some of its unique features, as well as the main research areas that are directly related to the further development of EVAR.

One of our most ambitious goals is to merge our approach to the integrated recognition of words and phrase boundaries with that of stochastic semantic analysis. The classification of words, phrase boundaries, and semantic attributes will then be subject to a single integrated search procedure, which employs many different sources of information, such as acoustics, prosody, statistical language models for word and boundary sequences, statistical models for semantic attributes, and statistical language models for sequences thereof. By replacing the second pass of the word recognizer with this search procedure, all these sources of information can be made available at an early stage in the recognition process. The result of this search will be an '*interpretation graph*' which contains not only words and syntactic boundaries, but also semantic interpretations of word sequences.

10. REFERENCES

1. M. Aretoulaki, S. Harbeck, F. Gallwitz, E. Nöth, H. Niemann, J. Ivanecky, I. Ipsic, N. Pavesic, and V. Matousek. SQEL: A Multilingual and Multifunctional Dialogue System. In *Int. Conf. on Spoken Language Processing*, Sydney, 1998.
2. M. Aretoulaki, S. Harbeck, E. Nöth, H. Niemann, I. Ipsic, N. Pavesic, J. Netrvalova, V. Matousek, J. Ivanecky, and Krokavec. A Multilingual and Multifunctional Spoken Dialogue System for Travel Information Access. *Speech Communication*, (To appear).
3. M. Boros, M. Aretoulaki, F. Gallwitz, E. Nöth, and H. Niemann. Semantic Processing of Out-Of-Vocabulary Words in a Spoken Dialogue System. In *Proc. European Conf. on Speech Communication and Technology*, volume 4, pages 1887–1890, Rhodes, 1997.
4. M. Boros, J. Haas, V. Warnke, E. Nöth, and H. Niemann. How Statistics and Prosody can guide a Chunky Parser. In *Proc. of the AIII Workshop on Artificial Intelligence in Industry*, pages 388–398, Stara Lesna, Slovakia, 1998.
5. W. Eckert. *Gesprochener Mensch–Maschine–Dialog*. Berichte aus der Informatik. Shaker Verlag, Aachen, 1996.
6. C. Féry. *German Intonational Patterns*. Niemeyer, Tübingen, 1993.
7. F. Gallwitz, A. Batliner, J. Buckow, R. Huber, H. Niemann, and E. Nöth. Integrated Recognition of Words and Phrase Boundaries. In *Int. Conf. on Spoken Language Processing*, Sydney, 1998.
8. F. Gallwitz, E. Nöth, and H. Niemann. A Category Based Approach for Recognition of Out-of-Vocabulary Words. In *Int. Conf. on Spoken Language Processing*, volume 1, pages 228–231, Philadelphia, 1996.
9. F. Gallwitz, E. Schukat-Talamazzini, and H. Niemann. Integrating Large Context Language Models into a Real Time Word Recognizer. In N. Pavesic and H. Niemann, editors, *3rd Slovenian–German and 2nd SDRV Workshop*, pages 105–114. Faculty of Electrical and Computer Engineering, University of Ljubljana, Ljubljana, 1996.
10. J. Haas, J. Hornegger, R. Huber, and H. Niemann. Probabilistic semantic analysis of speech. In E. Paulus and F. M. Wahl, editors, *Mustererkennung 1997, DAGM–Symposium*, pages 270–277, 1997.
11. J. Haas, E. Nöth, and H. Niemann. Semantigrams – Polygrams Detecting Meaning. In *Proc. of the 2nd SQEL Workshop on Multi-Lingual Information Retrieval Dialogs*, pages 65–70, Pilsen, 1997. University of West Bohemia.
12. R. Huber, E. Nöth, A. Batliner, J. Buckow, V. Warnke, and H. Niemann. You BEEP Machine — Emotion in Automatic Speech Understanding Systems. In *Proc. of the Workshop on TEXT, SPEECH and DIALOG (TSD'98)*, pages 223–228, Brno, 1998. Masaryk University.
13. R. Kompe. *Prosody in Speech Understanding Systems*. Lecture Notes for Artificial Intelligence. Springer–Verlag, Berlin, 1997.
14. R. Kompe, A. Kießling, H. Niemann, E. Nöth, A. Batliner, S. Schachtl, T. Ruland, and H. Block. Improving Parsing of Spontaneous Speech with the Help of Prosodic Boundaries. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, volume 2, pages 811–814, München, 1997.
15. S. McGlashan, F. Andry, and G. Niedermair. A proposal for sil. Technical report, University of Surrey, CAP SOGETI and Siemens AG, 1991. ESPRIT Projekt P2218 - SUNDIAL.
16. K. Mecklenburg, P. Heisterkamp, and G. Hanrieder. A robust parser for continuous spoken language using prolog. In *Proceedings of the Workshop on Natural Language Understanding and Logic Programming (NLULP 95)*, pages 127–141, Lissabon, 1995.
17. H. Niemann, E. Nöth, A. Kießling, R. Kompe, and A. Batliner. Prosodic Processing and its use in Verbmobil. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, volume 1, pages 75–78, München, 1997.
18. E. Schukat-Talamazzini, F. Gallwitz, S. Harbeck, and V. Warnke. Rational Interpolation of Maximum Likelihood Predictors in Stochastic Language Modeling. In *Proc. European Conf. on Speech Communication and Technology*, volume 5, pages 2731–2734, Rhodes, Greece, 1997.
19. E. Schukat-Talamazzini, T. Kuhn, and H. Niemann. Speech Recognition for Spoken Dialogue Systems. In H. Niemann, R. De Mori, and G. Hanrieder, editors, *Progress and Prospects of Speech Research and Technology: Proc. of the CRIM / FORWISS Workshop*, PAI 1, pages 110–120, Sankt Augustin, 1994. Infix.
20. M. Steedman. Grammar, Intonation and Discourse Information. In G. Görz, editor, *KONVENS 92*, Informatik aktuell, pages 21–28. Springer–Verlag, Berlin, 1992.
21. B. Tischer. *Die vokale Kommunikation von Gefühlen*, volume 18 of *Fortschritte der psychologischen Forschung*. Psychologie Verlags Union, Weinheim, 1993.
22. W. Wahlster. Verbmobil — Translation of Face–To–Face Dialogs. In *Proc. European Conf. on Speech Communication and Technology*, volume "Opening and Plenary Sessions", pages 29–38, Berlin, 1993.
23. C. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. Price. Segmental Durations in the Vicinity of Prosodic Boundaries. *Journal of the Acoustic Society of America*, 91:1707–1717, 1992.
24. H. Zeevat. Combining categorial grammar and unification. In R. T. Oehrle and U. Reyle, editors, *Natural Language Parsing and Linguistic Theories*, pages 202–229. D. Reidel Publishing, Dordrecht, Boston, 1988.