# Issues in the Evaluation of Spoken Dialogue Systems - Experience from the ACCeSS Project

# Thomas Brey\*, Gerhard Hanrieder<sup>†</sup>, Paul Heisterkamp<sup>#</sup>, Ludwig Hitzenberger\*, Peter Regel-Brietzmann<sup>#</sup>

\* University of Regensburg Universitätsstr. 31 D-93040 Regensburg Germany Thomas.Brey@ sprachlit.uni-regensburg.de Ludwig.Hitzenberger@ sprachlit.uni-regensburg.de

<sup>†</sup>Temic Speech Processing Soeflinger Strasse 100 D-89077 Ulm Germany Gerhard.Hanrieder@temic.com \*DaimlerChrysler AG Wilhelm-Runge Strasse D-89081 Ulm Germany paul.heisterkamp@ daimlerchrysler.com peter.regel-brietzmann@ daimlerchrysler.com

Abstract

We describe the framework and present detailed results of an evaluation of 1.500 dialogues recorded during a three-months field-trial of the ACCeSS Dialogue System. The system was routing incoming calls to agents of a call-center and handled about 100 calls per day.

# 1. Introduction

The ACCeSS Project aimed at offering flexible and robust speech interaction for call center automation. The prototypes were developed for different users, insurance services, call center services and call routing. A long time evaluation over three months in a production environment of one of the users showed the feasibility of complex dialogue systems.

# 2. System Description

This section describes the evaluated system in terms of dialogue functionality and system architecture.

# 2.1. System Functionality

The domain of the system is call routing for the support hotline of a pc manufacturer. The main task of the system is to transfer incoming calls to the responsible callcenter agent, i.e. the goal of the dialogue is to find out the reason for a call and initiate a call transfer directly to the most appropriate agent group. Such an automated preselection of calls can improve the efficiency of human operators to a great extent by keeping away routine transfer tasks from them.

The requirements analysis carried out with the hotline provider lead to a structuring of the dialogue into the following subdialogues:

- Language Selection: Since both German and English speaking customers call the hotline, an initial bilingual prompt was designed to choose the desired language. Callers wanting English support were directly transferred to an English speaking agent, i.e. the rest of the dialogue was implemented only in German.
- Device Selection: The task of this subdialogue was to find out whether the call was due to problems with a notebook, a desktop PC or any other device. In the latter case, callers were directly transferred to the call center responsible for non-PC devices.

- Concern of Call: In this subdialogue callers were asked whether they wanted technical support, product information, or register for warranty. In the latter two cases they were directly transferred to the information and warranty hotline, respectively.
- On-site service. Callers wanting technical support for a desktop PC were asked whether they needed the onsite service. If yes, they were asked for a 4-digit registration number and transferred depending on the validity of the number. If they had not yet registered for the on-site service, they were transferred to a different call center for prior registration. Since the on-site case with authentification is the menu with most depth, we provided callers at the end of this subdialogue with a shortcut, which they can speak in the beginning of their next call and are then directly asked for their registration number.
- Type of User: Callers wanting technical support for a notebook or desktop (no onsite-service) were asked whether they were dealers or end-users and transferred accordingly.

# 2.2. System Architecture

The demonstration system which was installed in the call-center environment of Telcare was a combination of a commercial speech computer platform with ACCeSS components. It was planned from the beginning of the project that the final system deployment should be done by an external integrator with special knowledge in the domain of computer telephony integration. Accordingly, the system design put much emphasis on a modular approach to ease the integration of the ACCeSS components into commercial IVR platforms (cf. Hanrieder: 98).

The chosen IVR platform was the PC-based Teamstar system (running under SCO Unix) by 4COM. The ACCeSS components speech recognizer and dialogue manager were integrated with the Teamstar system as follows: the ACCeSS components were running on a separate PC under Windows NT. This NT PC was connected to the Teamstar system via TCP/IP. The Teamstar system itself was connected to the telephone environment (a Meridian by Northern Telecom) and received speech data through a line interface board. Speech data were transferred via TCP/IP to the NT PC where speech recognition took place. Recognition results were then handed over to the dialogue module also running on the NT PC. The dialogue results were then processed by the control process running on the Teamstar platform. This process was responsible for handling multiple lines, playing appropriate speech files, and finally transferring the calls to human operators. Figure 1 illustrates this architecture.



Figure 1. System Architecture

The Teamstar system can handle 30 lines, but since in the call-routing domain each incoming call requires an outgoing line, the number of parallel calls was restricted to 15. The system characteristics were as follows:

Teamstar PC-based system with Intel Celeron 366 MHz Processor, 256 MB Ram, 10/100 Mbit Ethernet, Dialogic D300, S2m-connection with DSS1 protocol, SCO Open Server 5.05. The Windows NT PC was a Toshiba Equium Pentium III (450 Mhz) with 256 MB Ram.

The speech recognizer used was TEMIC's product version StarRec DSR. The dialogue manager was the prototypical Prolog implementation developed by TEMIC within the ACCeSS project (cf. Hanrieder, 1998).

# 3. Evaluation Framework

## 3.1. Methodology and Metrics

In (Hanrieder, Heisterkamp, Brey, 1998) we already described the set of metrics used for glass-box and black-box evaluation, which are with few exceptions part of the EAGLES standard (cf. Frazer, 1997). Here we simply list the standard metrics and limit the discussion to the extensions we made.

## 3.1.1. Glass-Box Evaluation

Since we used a phrase spotting version of the recognizer, word and sentence accuracy seemed not to be approriate measures for recognizer performance. By spotting only those words and expressions which contribute to the semantic content of the utterance, word accuracy can be bad, although the semantic content or

underlying concept of the utterance is perfectly understood:

System: Do you need product information, registration for warranty or technical support?

| User:   | Yes  | I need technic | al support |
|---------|------|----------------|------------|
| Recogni | zer: | technical      | support    |

We therefore decided to restrict glass-box evaluation to the evaluation of *concept accuracy* (CA) (cf. Boros et. al., 1996 and 1998). CA is determined as follows:

During runtime, the system logs for each turn the spoken user input in pcm format, recognizer output and the current dialogue context. After all spoken user input has been transcribed, we can automatically compare for each turn the result of semantic analysis of recognizer output with the result of semantic analysis of the transcribed string (in a given dialogue context). For the example above, our partial parser would produce the same output for recognizer result and transcribed user input. CA then tells us for a given sample the percentage of all user turns, where "the correct meaning" was computed from speech recognition result.

#### **3.1.2.** Black-Box Evaluation

The main extension to the EAGLES metrics was the replacement of Transaction Success Rate (TSR) as a measure of overall system performance, by a more finegrained metrics we call Subtask Success Rate (SSR). Instead of determining overall success or failure of a whole dialogue, we identify for a given application a set of subtasks the user may or has to accomplish in course of a dialogue. Motivation for this approach was the complex task structure of the application we described in (1998), which makes application of TSR very difficult. Because the user's intention may change during a single session or because a single "main goal" for which success or failure could be determined simply may not exist on the user's side, TSR seemed not to be appropriate. Decision about success or failure can easier be made regarding a given dialogue at the subdialogue or subtask level, resp. (e.g. language selection, device selection). SSR for a subtask can then be defined as the percentage of successful attempts of the user to complete this subtask. Note that the meaning of "attempt" in this definition does not include the correction of misunderstandings: An attempt may be successfully completed after n correction steps or not, but it remains one single attempt.

Applying SSR also in the simpler case of the call routing application<sup>1</sup>, again shows its diagnostic benefits, because it indicates not only that there are weak points in system design, but also where they are. For the current evaluation, the underlying set of subtasks (see below, Table 4) mainly correspond to the subdialogues described in Sect. 2.

In addition to SSR, we used the following standard metrics:

- Number of Turns
- Turn Duration
- Dialogue Duration
- Correction rate, which is defined as the percentage of all turns, whose purpose is only to clarify misunderstandings (cf. Frazer, 1997)

Since Dialogue Duration is included in the standard metrics anyway, using SSR allows the introduction of another metric *efficiency*, which can then be defined as the number of subtasks successfully completed in a given time interval.

Determining SSR requires detailed inspection of the logfiles by the evaluator, because success can only be measured given corresponding attempts on the user's side to accomplish the subtask in question. For which subtasks the user actually made attempts and how many, can only be decided by manually looking at what the user actually said. But whereas pure transcription as it is neccessary for glass-box evaluation anyway, only transfers spoken words to written language, human effort in determining SSR lies in the *interpretation* of the data. Thus we can not automatize this part of evaluation, but can only support it as much as possible.

If using SSR needs human interpretion, you encounter always the problem of the objectivity or at least intersubjectivity of the approach. To invalidate this argument, a sample of 300 dialogues within the field trial was evaluated by three different persons and the deviation of the resulting SSRs from the average SSR was less than 2%.

## 3.2. Software-Support

Most parts of the evaluation procedure make use of the logfiles, the system records automatically for each session. The logfiles contain the recorded speech input in pcm format and input/output protocols listing the system output and the recognized user input. For the evaluated samples of our field trial this yields to a total amount of about 45.000 files. As a first step, we developed a tool which generates for easy exploitation a HTML based hypermedia structure linking all logfiles together. Templates for transcription and extended input/output protocols are also created at this stage. Thus, even large corpora can easily be browsed without knowledge of the format of the logfiles. For the human annotation of dialogues needed for black-box evaluation, we provided the evaluators with an easy-to-use interface, which also graphically represents the structure of the corpus in temporal order.

Moreover, the tool performs all parts of evaluation that can be done automatically and it integrates other existing tools, e.g. for the evaluation of concept accuracy. The tool provides a search function, which enables the evaluator to find dialogues in a sample, matching certain criteria concerning the current set of evaluation metrics, e.g. to find dialogues with a high correction rate or with a low SSR for a certain subtask. After all the evaluation is done, the tool generates a report containing all values for the used metrics. Any files, the tool generates, are linked within the initial hypermedia structure, so that in the end, the corpus together with all annotations and evaluation results can easily be accessed and distributed. The complete, annotated corpus of the current evaluation contained in the end over 80.000 files which can all be accessed using an ordinary HTML-Browser.

<sup>&</sup>lt;sup>1</sup> Multiple attempts for a subtask are not possible in this case.

## 4. Results from the Field Trial

Equipped with methodology and tool support described so far, we evaluated 1528 dialogues, 1147 in the beginning (Phase 1) and 381 in the end of the field-trial (Phase 2). A characterization of the corpus is given in the following table:

| Metrics                             | Phase 1 | Phase 2 |
|-------------------------------------|---------|---------|
| Recorded Utterances                 | 6950    | 2209    |
| Total Words (Tokens)                | 10030   | 2856    |
| Total Expressions (Tokens)          | 6372    | 2107    |
| Distinct Words (Types)              | 174     | 170     |
| Distinct Expressions (Types)        | 200     | 163     |
|                                     |         |         |
| $\varnothing$ Number of Turns       | 15,2    | 14,1    |
| $\emptyset$ Dialogue Duration (sec) | 63,9    | 70,3    |
| Ø Turn Duration (sec)               | 4,1     | 4,6     |
| Total Dialogue Duration (min)       | 1222    | 416     |

#### Table 1. Corpus Characterization

The following table contains the results from glass-box evaluation:

| Metrics               | Phase 1 | Phase 2 |
|-----------------------|---------|---------|
| Concept Accuracy (CA) | 89,73 % | 90,40 % |

Table 2. Concept Accuracy

The following table contains the results from correction analysis. Some calls, especially those where the caller hangs up at the very beginning were not evaluated.

| Metrics              | Phase 1 | Phase 2 |
|----------------------|---------|---------|
| Correction Rate (CR) | 4,0 %   | 5,1 %   |
| Not evaluated        | 9,8 %   | 12,3 %  |

#### Table 3. Correction Analysis

Finally we present the results from subtask evaluation. As noted above, we hereby also have to take into account the number of attempts. In the following table we therefore also include the percentage of dialogues, in which no attempt for a subtask was made at all. In calculating an average success rate for all subtasks, we weight the single success rates with the percentage of attempts undertaken for each subtask. If e.g. we have attempts for subtask s1 in 80% of all dialogues and for subtask s2 in only 30% of all dialogues, an average SSR for s1 of 90% and an average SSR for s2 of 50%, an average Success Rate (SR) can then be obtained as follows:

$$SR = (90.0.8 + 50.0.3) / 1.1 = 79,1 \%$$

The following table contains the result of Subtask Analysis:

| Subtask                     | Phase 1  | Phase 2  |
|-----------------------------|----------|----------|
| Help                        |          |          |
| Success Rate                | 90,90 %  | 100,00 % |
| No Attempt                  | 77,40 %  | 80,31 %  |
| Device                      |          |          |
| Success Rate                | 98,44 %  | 100,00 % |
| No Attempt                  | 1,62 %   | 3,94 %   |
| Concern                     |          |          |
| Success Rate                | 99,70 %  | 100,00 % |
| No Attempt                  | 10,40 %  | 11,29 %  |
| User-Type                   |          |          |
| Success Rate                | 99,72 %  | 100,00 % |
| No Attempt                  | 23,93 %  | 22,57 %  |
| Registration Number         |          |          |
| Success Rate                | 72,29 %  | 70,00 %  |
| No Attempt                  | 86,83 %  | 82,15 %  |
| Language Selection          |          |          |
| Success Rate                | 68,46 %  | 100,00 % |
| No Attempt                  | 87,78 %  | 86,35 %  |
| On-Site Shortcut            |          |          |
| Success Rate                | 100,00 % | 100,00 % |
| No Attempt                  | 89,63 %  | 83,99 %  |
|                             |          |          |
| Average Success Rate (SR)   | 96,40 %  | 98,37 %  |
| Efficiency (subtasks / min) | 2,5      | 2,3      |

#### Table 4: Subtask Analysis

And finally some further results, which may be of interest:

|                       | Phase 1 | Phase 2 |
|-----------------------|---------|---------|
| Hang-Ups              | 19,94 % | 12,60 % |
| Initial Hang-Ups      | 5,18 %  | 7,61 %  |
| No Operator availible | 44,66 % | 2,36 %  |

#### Table 5: Further Results

### 5. Interpretation of Results

Regarding Table 4 we see that digit entry needed for subtask *registration number* is a remaining weak-point of the system since we have rather low succes rates for both periods. On the other hand there seems to be no significant difference of performance between the two periods. Maybe we can regard the slight increase of SR as well as CR in combination with a decrease of hang-ups as an indicator for a better understanding of the system by the user at the end of the field-trial. People who called the system occasionally or frequently, maybe became more aware of the possibilities to correct misunderstandings and maybe were less confused than in the beginning.

We conclude this chapter with a note on a comparison of CA and SR. A CA-value of 90% is not bad for recognizer performance, but still far from being perfect. This does not mean that we cannot have successful and robust speech understanding systems using current recognizer technology, since SR is actually significantly higher than CA.

# 6. Conclusions

Detailed evaluation at a subtask level turned out to be applicable and fruitful for two systems with totally different application domains, if we also consider the evaluation of the former ACCeSS system. But whereas the former evaluation was restricted to laboratory tests, this time we could apply the method within a field trial of an integrated system.

Its diagnostic value turned out to be also of interest to the hotline provider when combined with online monitoring tools. If such a tool reports problems, detailed evaluation could show where they come from.

We finally remark that the hotline provider also made investigations in acceptance of the system and presented some callers with a questionnaire after the call. Most of them were quite content with the system. Only dealers critized the dialogue as too lengthy, they want directly be transfered to an agent. For this user group another shortcut facility as described in Sect.2. should be implemented.

## 7. References

- Hanrieder; G.: Integration of a mixed-initiative dialogue manager into commercial IVR platforms; In: Proceedings of IVTTA, Turin, 1998.
- Boros, M., Eckert, W., Gallwitz, F., Goerz, G., Hanrieder, G., Niemann, H.: *Towards Understanding Spontaneous Speech: Word Accuracy vs. Concept Accuracy.* In: Proceedings of ICSLP 1996, Philadelphia.
- Boros, M., Ehrlich, U., Heisterkamp P., Niemann H.: An Evaluation Framework for Spoken Language Processing, In: Proceedings of SPECOM 1998, St. Petersburg.
- Hanrieder, G., Heisterkamp, P. & Brey, T.: *Fly with the EAGLES: Evaluation of the ACCeSS Spoken Dialogue System*; In: Proceedings of ICSLP, Sydney, 1998.

Fraser, N.: Assessment of interactive systems.In: D. Gibbon, R. Moore, R. Winski (eds): Handbook of Standards and Resources for Spoken Language Systems; Berlin, 1997.