SpeechEval – Evaluating Spoken Dialog Systems by User Simulation

Tatjana Scheffler and **Roland Roller** and **Norbert Reithinger**
DFKI GmbH
Alt-Moabit 91c
10559 Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

In this paper, we introduce the SpeechEval system, a platform for the automatic evaluation of spoken dialog systems on the basis of learned user strategies. The increasing number of spoken dialog systems calls for efficient approaches for their development and testing. The goal of SpeechEval is the minimization of hand-crafted resources to maximize the portability of this evaluation environment across spoken dialog systems and domains. In this paper we discuss the architecture of SpeechEval, as well as the user simulation technique which allows us to learn general user strategies from a new corpus. We present this corpus, the VOICE Awards human-machine dialog corpus, and show how this corpus is used to semi-automatically extract the resources and knowledge bases on which SpeechEval is based.

1. Introduction

The more spoken dialog systems (SDSs) are put into practice in different domains, the more efficient methods for their development and deployment are urgently needed. The project SpeechEval aims to address this need in two ways: First, by investigating the use of dialog corpora in order to automatically or semi-automatically create the resources necessary for the construction of SDSs. And second, by learning general user behavior from the same corpora, and building a flexible user simulation which can be used to test the overall usability of SDSs during development or after deployment.

Automatic testing of dialog systems is attractive because of its efficiency and cost-effectiveness. However, previous work in this area concentrated on detailed tests of individual subcomponents of the SDS (such as the ASR), or on small systems in toy domains. In order to judge the overall usability of a commercial dialog system, extended testing by human callers has been necessary – a step that is usually too costly to be undertaken during the prototype stage or repeatedly after changes to the deployed system. SpeechEval intends to fill this gap, providing a flexible user simulation platform which allows automatic repeated testing of an SDS. Maximum modularity of the system architecture as well as the automatic and semi-automatic techniques for the creation of the underlying resources for the user simulation (in particular, domain knowledge and user strategies) allow SpeechEval to be easily portable across different SDSs.

In this paper, we concentrate first on the user simulation technique in SpeechEval. Then we describe the architecture of the SpeechEval platform. We pay special attention to the resources (general, domain- or system-dependent) which need to be constructed or adapted when using SpeechEval as a user simulation for a new application. The rest of the paper describes our finished and ongoing work in extracting knowledge bases for the SpeechEval system from corpora.

2. User Simulation

User simulation is used in the SDS literature for several purposes. First, for training the dialog manager of a spoken dialog system during reinforcement learning. In this case, the SDS with the learned strategy is the actual purpose of the research, whereas the user simulation is just a means to that end. Second, user simulation is used for evaluation or testing of the trained policies/dialog managers of the developed spoken dialog systems. The two types of purposes of user simulations may call for different methods. A good overview of state-of-the-art user models for SDS training is given in (Schatzmann et al. 2006). A user simulation may be used to test for general soundness of an SDS, specifically searching for errors in the design. In such a case, a random exploration may be called for (Alexandersson and Heisterkamp 2000). A restricted random model may also perform well for learning (Ai, Litman, and Litman 2007).

In other cases, ideal users may be modelled so that reinforcement learning is able to learn good paths through the system's states to the goal (López-Cózar et al. 2003). In an approach closer to our work, (Chung 2004) developed a variable user simulation used for detecting potential errors in a SDS with a large database back-end. In both projects, the user simulation is hand-crafted by the designer of the SDS.

Our goal in SpeechEval is to as much as possible avoid hand-crafting the strategy (i.e., user simulation). Since in our case the user simulation itself is the goal and not merely a step along the way, the requirements for the user model may also differ from previous approaches. An optimal strategy is not needed for our user simulation, neither is a random explorative strategy. Instead, the aim should be *realistic* user behavior. Since SpeechEval should be used to evaluate spoken dialog systems in parallel or instead of human judges, it should show similar behavior (at least asymptotically) to these judges. The behavior of human evaluators of spoken dialog systems can be observed in our corpus, the VOICE Awards Corpus described below in section 4. We therefore define realistic user behavior in our case as user utterances that probabilistically match the ones represented in our corpus. Such probabilistic models are often used for evaluation of learned dialog managers (Ai, Litman, and Litman 2007). How to effectively measure the realism of simulated dialogs is still very much an open research question. Some measures are discussed for example in (Jung et al. 2009), based on comparing the simulated dialogs with real user dialogs using the BLEU metric and based on human judgments. In the absence of real user dialogs with the same SDS, we aim for greater variability in the simulated user behavior.

One method of achieving both greater realism and variability is the use of a true speech interface when interacting with the SDS to be evaluated. Previous work often reduces interaction to the text or even concept level, or uses canned user responses (as in the case of (López-Cózar et al. 2003)). In contrast, SpeechEval interacts with the SDS just like a human user would, over the telephone. The use of a text-to-speech system allows for greater variability in production than concept-based or canned output. It will allow us to tune the output and introduce disfluencies as well as errors and uncooperative behavior. On the other hand, using ASR and TTS modules obliviates the need to artificially "model" signal errors by introducing fixed error rates. Instead, errors will be introduced naturally through the normal telephone noise. The ASR component shows very good results so far, which should be able to match a human user as long as the ASR grammar is suitable. Furthermore, robust processing in the pipeline ensures that small ASR errors will not completely derail the response. Overall, the use of a real speech interface makes the simulated dialogs much more realistic and variable than it would otherwise be possible.

3. SpeechEval Architecture

The planned architecture of the SpeechEval system is shown in Figure 1. It essentially follows a standard pipelined architecture for spoken dialog systems, with some additional modifications to include the user simulation functionality. In this section, we briefly describe the components of our system, and the resources which are necessary to use SpeechEval to evaluate a given SDS. Such resources may be general, domain- or system-specific. We discuss in each case, whether they must be specified by hand or can be learned (and how).

SpeechEval will be implemented using the Ontology-based Dialogue Platform (ODP) a generic modeling framework and run-time environment for multimodal dialog applications. For a more detailed description, see (Pfalzgraf et al. 2008).

There are three central knowledge bases which need to be acquired off-line before launching the system: (1) A domain ontology, which contains domain-specific information about available objects and actions and must be specified by hand. SpeechEval provides functionality which supports and speeds up the construction of this ontology. (2) A set of user goals to be used during the user simulation. Such goals are highly system and domain specific and must be specified by a domain expert. This goal set is equivalent to the instructions provided to human testers and therefore does not in itself constitute a significant impediment of using SpeechEval for automatic testing. (3) A user stereotype. Possible user characteristics are extracted from a dialog corpus (see below). SpeechEval allows testing with different user characteristics (such as fast or slow reaction time, many/few barge-ins, or differing error rates). A GUI is planned which allows the SpeechEval user to set these characteristics in an intuitive way.

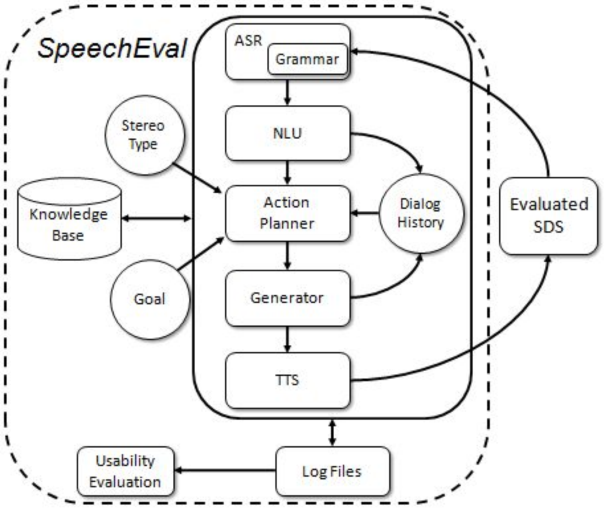

Figure 1: Architecture of the SpeechEval system.

During on-line runs, SpeechEval's architecture largely follows a standard pipeline model. The speech signal with the SDS prompt received via telephone is first processed in the ASR component. The recognition grammar is learned in a multi-step process using our human-machine dialog corpus (introduced in section 4) as well as other sources. This obliviates the need for tedious hand-tuning of the grammar, and thus makes SpeechEval much more easily portable to new SDSs and domains.

The second step of natural language understanding (NLU) consists of three parts. The segmentation and dialog act classification components are learned from our annotated corpus. We follow the approach in the AMIDA project (AMIDA 2007) for the segmentation. For the dialog act classification, we use a Maximum Entropy classifier trained with the WEKA toolkit (Witten and Frank 2005). Our implementation is based on the work by (Germesin 2008). However, in an on-line system such as SpeechEval, features based on future dialog act assignments cannot be used. The third component of the NLU module performs a keyword search and other information retrieval steps to link the incoming prompt to the domain ontology.

The action planner is the central step in the pipeline. Based on the analysis of the incoming prompt, a reply action is devised. Our current target approach is very close to the one proposed in (Georgila, Henderson, and Lemon 2005) for an information state update system. At each state in the dialog, the user model choses the next action based on the transition probabilities observed in the corpus. Since some states have never or only rarely been seen in the corpus, we choose a vector of features as the representation of each dialog state. These features in our case include properties of the dialog history (such as the previous dialog act, the number of errors), the current user characteristics (expert vs. novice, for example), as well as other features such as the ASR confidence score. We estimate from the corpus the amount that each feature in the vector contributes to the choice of the next action. Thus, unseen states can be easily mapped onto the known state space as they lead to similar behavior as closely related seen states would.

The chosen next action is a dialog act type that must be enriched with content based on the goal and user characteristics. General heuristics are used to perform this operation of tying in the user simulation with the domain- and system-specific ontology.

The output of the action planner is an utterance plan including a dialog act type and additional information. This is used in the generator to produce an answer string of the user simulation. The generator follows a simple template-based approach. Our corpus shows that by far the largest part of user turns in commercially deployed spoken dialog systems consist of just a single word. Thus, a very simple baseline generator just outputting single words or short phrases (e.g., number sequences) seems sufficient for reasonably realistic generation. In the future, we intend to extract templates of longer user utterances from the corpus in order to improve on the generator's performance and in order to make its output more variable for testing purposes.

An out-of-the-box text-to-speech system is used to render the generated utterances in spoken German, which is then sent on to the SDS per telephone.

The actual usability evaluation of the SDS is performed in a separate module that can keep track of the incoming utterances and their analysis, as well as the outgoing messages and internal state (e.g., the current user characteristics). The evaluation is based only on objective measures like dialog act types, turn durations, etc. and data derived from these measures, since user judgments as for example in the PARADISE evaluation metric (Walker, Kamm, and Litman 2000) cannot be obtained. The details of this usability evaluation are not the focus of this paper, however.

4. A Human-Machine Dialog Corpus

Development of spoken dialog systems takes time, because the rules and knowledge bases for a new system must be acquired in one of two ways: In a hand-crafted system, which includes virtually all current commercially deployed systems, all rules and knowledge bases must be specified by a human expert. This requires expert knowledge by the designer not only of the underlying dialog platform and architecture, but also about the content domain and interaction structure of the planned dialog system. As an alternative to hand-crafted systems, the strategies in a SDS may be learned automatically from available corpora. Much research has been done in this area recently, especially on dialog strategy optimization by reinforcement learning with (Partially Observable) Markov Decision Processes ((PO)MDPs) (see for example (Lemon and Pietquin 2007) for an overview). This approach works best for learning very specific decisions such as whether or not to ask a confirmation question or how many pieces of information to present to a user (Rieser and Lemon 2007). In addition, such systems must have access to large corpora of interactions with the particular system for training, creating a chicken-and-egg problem. The goal of SpeechEval, however, is to be able to interact with a new SDS in a new domain with little modification. In particular, SpeechEval should be able to evaluate a prototype SDS for which no specialized corpus of human-SDS interactions exists. Therefore, we aim to learn general strategies of user behavior as well as other kinds of knowledge bases for the SpeechEval system from a general dialog corpus.

Since we could not identify an appropriate human-machine dialog corpus in German, we are currently in the process of compiling and annotating the VOICE Awards corpus, which will be a large collection of recordings of dialogs with SDSs from all possible commercially deployed domains. It is based on the "VOICE Awards" competition of German language SDSs.

The annual competition "VOICE Awards"[1] is an evaluation of commercially deployed spoken dialog systems from the German speaking area. Since 2004, the best German spoken dialog applications are entered in this benchmarking evaluation, where they are tested by lay and expert users. We are currently in the process of constructing an annotated corpus of the available audio recordings from this competition, including the years 2005–2008.

The corpus represents a large breadth of dialog systems and constitutes a cut through the current state-of-the-art in commercially deployed German SDSs. Altogether, there are 130 dialog systems in the corpus, with about 1900 dialogs. In each year of the competition, 10 lay users were asked to call the dialog systems to be tested and perform a given task in each of them. The task was pre-determined by the competition organizers according to the developers' system descriptions, and these tasks are usually the same for all 10 lay users. After completing the task, the users filled out satisfaction surveys which comprised the bulk of the evaluation for the award. In addition to these lay callers, two experts interacted with each system and performed more intensive tests, specifically to judge the system's reaction to barge-ins, nonsensical input, etc. These interactions are only in some cases included in the corpus. Table 1 contains a list of some of the domains represented by the dialog systems included in the VOICE Awards corpus.

Audio data for the VOICE Award corpus is available in separate .wav files for each dialog. The transcription of the corpus, using the open source Transcriber tool[2], is about 50% complete. With the transcription, a rough segmentation into turns and dialog act segments is being performed. Since more fine-grained manual timing information is very

[1] http://www.voiceaward.de/

[2] http://trans.sourceforge.net/

public transit schedule information
banking
hotel booking
flight info confirmation
phone provider customer service
movie ticket reservation
package tracking
product purchasing

Table 1: Some domains of SDSs included in the VOICE Awards corpus.

difficult and time-consuming to obtain, it is planned to retrieve word-level timing by running a speech recognizer in forced alignment mode after the transcription is completed.

As a basis of our statistical analyses, the entire corpus is being hand-annotated with several layers of information: (1) Dialog acts, (2) sources of miscommunication, (3) repetitions, and (4) task success. Since the lack of space prohibits a detailed discussion, the annotation schemas are simply listed in table 2. We are using a modified tool from the NITE XML Toolkit (NXT)[3] that has been adapted to our needs to perform these annotations in a single step. The result will be a large corpus of human-SDS-dialogs from many different domains, covering the entire breadth of the current state-of-the-art in commercially deployed German SDSs.

Several other layers of annotation will be added automatically for purposes of saving time, error reduction and consistency. This includes objective information that can be reliably estimated directly from the corpus, such as user reaction time, style and length of user utterances, etc. Some of these automatic annotations are listed in table 3.

5. Corpus-Assisted Creation of SDS Resources

As one of the major goals of the SpeechEval systems is easy portability across systems (to be evaluated) and domains, many of the knowledge bases and resources must be learned from corpora. The main corpus for our development is the VOICE Awards corpus described above, which presents a cross-section through many current SDSs. In this section, we describe how this corpus is being used, along with some supplementary sources, to derive the knowledge bases that are part of the SpeechEval architecture (see section 3).

ASR Grammar

In order to improve the coverage of SpeechEval's speech recognition, the recognizer's grammar must be augmented by adding both domain specific terminology as well as terms and phrases that are important in the scenario of spoken dialog systems in general. Different strategies will be used to extract both kinds of vocabulary from the VOICE Awards Corpus as well as other sources.

dialog acts	hello
	bye
	thank
	sorry
	open_question
	request_info
	alternative_question
	yes_no_question
	explicit_confirm
	implicit_confirm
	instruction
	repeat_please
	request_instruction
	provide_info
	accept
	reject
	noise
	other_da
miscommunication	not_understand
	misunderstand
	state_error
	bad_input
	no_input
	self_correct
	system_command
	other_error
repetition	repeat_prompt
	repeat_answer
task_success	task_completed
	subtask_completed
	system_abort
	user_abort
	escalated
	abort_subtask

Table 2: Hand-annotation schemas of the VOICE Awards corpus.

For the extraction of domain specific terminology, we have categorized the systems in the corpus into domains. A simple chi-square test is used to determine whether a certain word i is significant for a domain j: Given the number of times i occured in j (O_{ij}) and the expected frequency of i in j according to the distribution in the entire corpus (E_{ij}), the chi-square value of the word i for the domain j is computed using the following formula:

$$\chi^2 = \Sigma_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad (1)$$

where the expected frequencies E_{ij} are computed using the following occurrence counts, and formula 2:

	domain j	$\neg$ domain j
word i	a	b
$\neg$ word i	c	d

$$E_{ij} = \frac{(a + c) \times (a + b)}{(a + b + c + d)} \quad (2)$$

Using a stop-word list of the 1000 most frequent terms in German, any word with a chi-square value greater than 3.84 (and whose observed count is higher than the expected one) is likely ($p < 0.05$) to be significant for the domain. Words

[3] http://groups.inf.ed.ac.uk/nxt/

dialog length	time
length of turns	time
# dialog turns	# interactions # sds prompts # user turns
user reaction time	(by forced alignment)
style of user utterance	single word phrase full sentence SDS-specified / free option

Table 3: Automatic annotations of the VOICE Awards corpus.

which occurred less than 5 times in the corpus were discarded since the test is likely to be inaccurate. This method yielded very good results even when evaluated on a very small subcorpus. Table 4 shows the top 15 positively significant words for the banking domain, as computed on only 58 dialogs from the domain, and a similar amount of out-of-domain dialogs. The only false hits are words that are very suggestive of customer service SDSs in general (e.g., “möchten” / “would like”). These can be excluded by a second stop word list, but they would also be very likely to disappear when a larger amount of data (i.e., the entire VOICE Awards corpus) is used in the computation.

term	**English**	χ^2
Kontostand	account balance	56.6
Kontonummer	account number	54.5
möchten	would like	44.1
Umsätze	transactions	40.7
Konto	account	40.2
Überweisung	wire transfer	32.9
Cent	Cent	29.1
minus	negative	28.1
Ziffer	digit	27.6
Geburtsdatum	birth date	26.0
Hauptmenü	main menu	23.9
Bankleitzahl	routing number	22.9
Servicewunsch	service request	21.8
beträgt	amounts to	21.3
Gutschrift	credit	20.8

Table 4: Significant words in the banking domain.

We plan on extracting SDS-specific terminology (such as “customer id”, “main menu”, etc.) using the same methodology. All dialogs in the VOICE Awards corpus can be used as the positive subcorpus. For the negative examples, we will use text extracted from web pages representing a similar range of topics and domains as the VOICE Awards corpus. This will ensure that only terminology specific to the medium of spoken dialog systems is marked significant by the chi-square test, and not other frequent content words such as domain-specific terms.

User Characteristics

In order to perform realistic testing of dialog systems, the user simulation’s behavior must be relatively varied. We aim to identify suitable user types from the VOICE Awards corpus to model them in our user simulation. Broad distinctions such as expert vs. novice users are known from the literature, but aren’t easily observable in the corpus, since by far most dialogs are by lay users. Thus, we instead try to distinguish objectively observable characteristics such as the user reaction time, number of barge-ins, etc. We will perform a clustering on each of these variables in order to obtain a “user properties vector” for each caller in the corpus. The obtained user characteristics then become part of the dialog state vector which determines the following user actions. This will account for the differences in behavior of different user types.

Dialog Act Segmentation and Classification

Machine learning approaches are the standard approaches to the tasks of dialog act segmentation of classification. Good results can be obtained when the number of classes is not too high, although the quality of the ASR output has a large impact on the accuracy, as well. In SpeechEval, we only distinguish 17 mutually exclusive dialog act types (see table 2). Further, the types can be grouped into a flat hierarchy of broad categories such as “question” and “answer”. Thus, even in cases where an incoming dialog act has been wrongly classified, SpeechEval’s reply may still be appropriate if the misclassified type is of the same super-category.

Our segmentation and classification follows closely the method developed in the AMIDA project (AMIDA 2007). We use the WEKA toolkit to implement separate segmentation and dialog act classification learners. As opposed to this previous work, we use the learned classification modules within an online system. This means that we cannot make use of dynamic features that require the knowledge of future assignments (as is done in the dialog act classifier). Each determined dialog act type is passed on immediately down the pipeline architecture and is acted upon in further modules. However, the reassignment of dialog act labels as done in the work of Germesin (2008) can be used in SpeechEval to retroactively change the dialog history. This may affect both the computation of later dialog act types as well as the confidence scores of SpeechEval’s replies.

User Utterance Templates

As noted above, by far most user utterances in our corpus consist of just one word. In an initial study, only 12% of the user turns contained more than one word (number sequences such as ID or telephone numbers were excluded). Most of these longer utterances were false starts or two-word names such as a person’s first and last name. Thus, a very simple user simulation baseline will just output the one word which constitutes the answer to the prompt.

For genuine more-word utterances, we are exploring a grammar induction technique in order to extract possible user utterance templates from our corpus. User utterances will be POS-tagged and the possible phrase structures are extracted. In order to find templates, we use our lists of domain-specific words as determined by the chi-square test described above. Domain words can thus be matched onto one another, and general templates with blanks can be extracted this way. The blank spaces are linked to the domain ontology. During generation, the blanks are filled from the ontology if such a template is chosen as a user utterance. With this method, even the rarer longer user utterances can be generated. The advantage is that the system designer does not have to hand-specify a list of possible user utterances in the domain. Instead, general templates are extracted which can be filled with domain vocabulary.

6. Hand-Specified Resources

Even though a goal of SpeechEval is the minimization of hand-crafted resources, certainly not everything can be automatized. In particular, a domain expert must specify the domain ontology which contains the available objects and relations in the domain. The automatically extracted domain vocabulary can be a basis of this ontology, but the relations must be specified by hand.

Further, the set of possible goals which SpeechEval is to pretend to solve must also be pre-specified. This is not surprising. In the VOICE Awards contest, the human judges are also given scenarios to solve for each system. The set of goals to be tested represents the scenario information for the computer evaluator (SpeechEval). During each dialog, one goal is chosen from the set.

7. Conclusion

In this paper we presented the SpeechEval system, a simulation environment that makes possible the quantification of the usability of spoken dialog systems with minimal use of human evaluators and hand-crafted resources. We presented SpeechEval's simple pipelined architecture, with a special focus on the necessary knowledge bases and resources.

In the second part of the paper, we introduced our corpus of German human-machine dialogs, which constitutes the basis of our statistical methods for extracting knowledge bases for spoken dialog systems. We discuss how most of the resources in the SpeechEval architecture, from the ASR grammar to dialog strategy, can be derived from the general dialog corpus or other supplementary corpora. This ensures easy portability of the SpeechEval user simulation across SDSs and domains.

We are currently integrating the system components and carrying out feasibility experiments. The full system will allow speedy evaluation of SDSs during development as well as after updates to deployed systems without the need for large specialized corpora or expensive human evaluators.

8. Acknowledgements

This work is part of the project "SpeechEval: Automatic Evaluation of Interactive Speech-Based Services on the Basis of Learned User Models", which is being carried out in cooperation with the Quality and Usability Lab at Technical University Berlin. SpeechEval is funded by the Investitionsbank Berlin through the ProFIT framework, grant #10140648. This project is being co-financed by the European Union (European Regional Development Fund).

We would like to thank the three anonymous reviewers for their detailed and helpful comments. All errors remain our own.

References

Ai, H.; Litman, T.; and Litman, D. 2007. Comparing user simulation models for dialog strategy learning. In *Proceedings of NAACL/HLT 2007*, 1–4.

Alexandersson, J., and Heisterkamp, P. 2000. Some notes on the complexity of dialogues. In *Proceedings of the 1st Sigdial Workshop on Discourse and Dialogue*, volume 10, 160–169.

AMIDA. 2007. Deliverable D5.2: Report on multimodal content abstraction. Technical report, DFKI GmbH. chapter 4.

Chung, G. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, 63–70.

Georgila, K.; Henderson, J.; and Lemon, O. 2005. Learning user simulations for information state update dialogue systems. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech).*

Germesin, S. 2008. Determining latency for on-line dialog act classification. In *MLMI'08.*

Jung, S.; Lee, C.; Kim, K.; Jeong, M.; and Lee, G. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech and Language* 23:479–509.

Lemon, O., and Pietquin, O. 2007. Machine learning for spoken dialogue systems. In *Proceedings of Interspeech.*

López-Cózar, R.; de la Torre, A.; Segura, J.; and Rubio, A. 2003. Assessment of dialog systems by means of a new simulation technique. *Speech Communication* 40:387–407.

Pfalzgraf, A.; Pfleger, N.; Schehl, J.; and Steigner, J. 2008. Odp: Ontology-based dialogue platform. Technical report, SemVox GmbH.

Rieser, V., and Lemon, O. 2007. Learning dialogue strategies for interactive database search. In *Proceedings of Interspeech.*

Schatzmann, J.; Weilhammer, K.; Stuttle, M.; and Young, S. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review.*

Schehl, J.; Pfalzgraf, A.; Pfleger, N.; and Steigner, J. 2008. The Babble-Tunes system - Talk to your iPod! In *Proceedings of the 10th international conference on Multimodal interfaces.*

Walker, M.; Kamm, C.; and Litman, D. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering* 6:363–377.

Witten, I., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco: Morgan Kaufmann, 2nd edition.