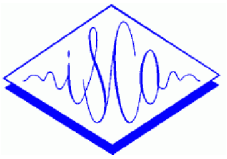
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Correlation Between Model-based Approximations of Grounding-related Cognition and User Judgments

Klaus-Peter Engelbrecht, Sebastian Möller

Quality and Usability Lab, Telekom Innovation Laboratories, TU Berlin, Germany
{Klaus-Peter.Engelbrecht, Sebastian.Moeller}@telekom.de

Abstract

As spoken dialog systems become more complex, efficient ways to evaluate them in early development stages are required. User simulation has been successfully used for this purpose. While current user models describe behavior on the level of overt behavior, modeling aspects of cognition can reveal direct insights into usability problems. Thus, in this paper we propose two models related to grounding in dialog: a model of the belief the user has about the system state, and a model of vocabulary alignment. We show that parameters derived from these models are significantly correlated with the users' quality perception.
Index Terms: user model, evaluation, grounding

1. Introduction

Due to the recent improvements in automatic speech recognition and natural language understanding, spoken dialog systems (SDS) have reached a degree of complexity which makes their design a challenge for developers. Thus, it has become increasingly important to evaluate the systems at design time to ensure that performance criteria are met and users are satisfied with the service. While methods exist to assess system quality with users, these are often used less than required, as they are demanding in terms of time, money and expertise [1]. Automatic evaluation has been proposed as a way to reduce the costs of user tests [2].

Evaluation of SDSs can be automized by carrying out user simulations [3, 4]. Such simulations require a user model, which produces a user action based on the previous system action and the dialog history. System actions are represented as dialog acts, which may be further specified using attributes (e.g. *query(food type)* or *no_match*). or attribute-value-pairs (AVPs) in case of confirmation acts (e.g., *confirm(food_type=Italian)*). User actions can be represented on the concept level [4, 5], in which case they can be described by AVPs, on the wording level [6], or on the utterance level [6, 7], depending on the system component to optimize.

Currently, most user models describe the overt behavior of users in a probabilistic manner [5, 8, 9] and do not consider cognitive processes inside the user. Previous research has shown that such simple user models are useful to detect design errors in an interface [9, 6], optimize different aspects of the system [7], or measure performance and predict user judgments [9, 10] in simulation experiments. However, it can be useful to also consider cognitive aspects of the interaction, such as the mental model the user has of a system.

While human cognition is very complex and not directly observable, we can make assumptions about certain aspects of it and analyze the validity of these assumptions based on observable user behavior. In this paper, we model two aspects of cognition which are both related to grounding in dialog: the belief the user has about the system state at each dialog exchange, and learning of the vocabulary and wordings the system can understand. Both aspects concern knowledge the user has about the system, and the proposed models describe how this knowledge is acquired in a dialog. Thus, they help analyzing if the system provided all information needed by the user to "know what to say" and correctly track the system state. This is an important aspect of SDS design, which so far had to be analyzed based on time-consuming expert annotations of log-files [11].

Explicitly representing grounding-related cognition in the user model may also allow a deeper integration with models for the prediction of their quality judgments. This is analyzed in more depth in this paper. User judgments have previously been predicted from interaction data using trained classifiers [12]. A main problem remains to find good predictors generalizing across different systems. The proposed models may yield parameters which are directly related to the dialog quality, and are thus more general predictors of user judgments.

In the following sections, we first describe our models for the believed system state and for learning correct expressions. Afterwards, we describe the dataset used as empirical basis for our study, which was collected with a mixed-initiative restaurant information system called BoRIS [13]. Finally, we explain how predictors of user judgments can be derived from the models and show via correlation analysis that the modeled aspects of cognition can be used to predict user ratings.

2. Belief Model

The believed system state is structured in the same way as the real system state. It consists of a set of slots for each type of input, e.g. in the case of BoRIS the price range, food type, location of the restaurant, date and time. In the system, these slots are filled with values provided in the user utterances. E.g., if the utterance "I want pizza" is observed, the system would add the canonical value "Italian" to the *food type* slot. Later, these values are used in the database query to find a matching restaurant. Contrary to the system state, the *believed* system state is not updated based solely on the AVPs mentioned by the user, but also based on the system feedback, e.g., in the simplest case, the confirmed AVPs.

Recent work circling around POMDP-based, self-learning SDSs has discussed how a system may track several concurring hypotheses about the previous user inputs in a probabilistic representation of the "believed" user tasks (e.g. 14). The main purpose of this is to cope with uncertainty in the understanding results. In contrast, the user will understand the system correctly in most cases, reducing the importance to track several possible believes. Our model is thus deterministic which greatly simplifies the implementation and avoids the problem of availability of sufficient training data.

The belief is updated from one exchange to the next using a set of update rules. Some example rules are presented in Table 1. These rules were defined based on intuition about what can be inferred from a system prompt. It can be seen that rules can refer to many different events in the dialog history and their interrelations. Astonishingly, however, for a human it is usually unambiguous which information the system has stored given the dialog history and the current prompt.

Processing these rules can be simplified by adding a new field to the system dialog act representation, which explicitly lists the information about the system state (or single slots) contained in the prompt. For example, in "Did you say *Italian* or *German* food?", the information *state.food = {Italian, German}* would be added.

Table 1. *Belief update rules (examples).*

Belief update rules (examples)
- Start with all slots empty. - In case of a no-match prompt, no change is required. - Add AVPs explicitly confirmed by the system. - In case affirmation of the confirmation by the user is required, and the user does not affirm or the system asks for any of the confirmed values in the next exchange, remove all confirmed values. - Clear slots queried by the system; however, if the system asks for repetition of the slot value, assume that the slot is filled with some value unknown to the user. - If the system provides no feedback on values specified by the user, add these values, but only if the system continues consistently (e.g. not asking for one of the provided slots)

3. Learning Wordings

In order to analyze how users learn wordings understandable to the system, we first need to separate the speech understanding (SU) errors annotated in the database into errors due to incorrect usage of vocabulary and expressions, and errors due to incorrect decoding of the speech signal into text. The frequency of the former errors is assumed to be impacted by the user's learning capabilities and the information the system provides about how to talk to it. The latter are assumed to be independent of this.

To determine the language-related errors in our database, we parse the expert transcriptions of the user utterances with the system's NLU component and compare the resulting AVPs to an expert annotation of the AVPs the user meant to express. Whenever a difference is found, the user used an expression which is not covered by the system's grammar. All other errors logged in the database are attributed to the speech decoding.

3.1. General approach to model understanding errors

Both types of errors can be modeled statistically using the same general approach. Like the entire interaction, they are modeled on the level of concepts, or AVPs, respectively. More specifically, given a user action consisting of several AVPs, understanding errors are modeled for each individual AVP, using a confusion matrix. The confusion matrix stores the conditional probabilities to observe each possible output AVP given a source AVP.

While traditionally SU models distinguish *deletions*, *substitutions* and *insertions*, we only differentiate between deletions and insertions, as substitutions can be modeled as deletion of an AVP and insertion of another one. By this, we release the assumption that some insertions can be explained uniquely by any of the AVPs in the source utterance. Rather, if an AVP a_j' is inserted, it is assumed that all AVPs in the utterance contribute to this equally. In contrast, correctly transmitted AVPs are assumed to be due to their counterpart in the source utterance.

Thus, the required confusion matrix can be learned from empirical data as follows: We count co-occurrences in a matrix $C = c_{ij}$, where c_{ij} is incremented by

- 1, if the AVP a_i appears in the utterance and a_j' in the SU result, and $i=j$.
- $1/N_{AVPs}$, if the AVP a_i is in the utterance, and a_j' is in the SU result, and $i \neq j'$, and a_j' cannot be explained by a_j being in the utterance. N_{AVPs} is the number of AVPs in the utterance.

Conditional probabilities $P(a_j' | a_i)$ are obtained by dividing each row i in C by the frequency of a_i in all source utterances. For the calculation of the error rates below, this normalization was inverted first. The rate of correctly understood AVPs can then be determined by dividing the diagonal elements of the matrix by the total number of AVPs mentioned by the users. The deletion rate is obtained by subtracting this number from 1. The insertion rate is determined by summing all non-diagonal elements of the matrix and dividing by the total number of mentioned AVPs.

3.2. Model of learning vocabulary and wordings

In order to analyze how users learn to use the right wordings, such error models were calculated for the AVPs satisfying (or not satisfying) one or several of the following factors:

- *Indicated AVPs*: concepts for which the system provided keywords in the current turn.
- *Successful AVPs*: concepts which the user thinks she has transmitted to the system successfully previously. We describe below how they are determined based on the belief model.
- *Known AVPs*: combination of all indicated and successful AVPs.
- *Asked slot*: The slot the system asked for in this exchange.

Table 2 shows that indeed users learned to use the right words. In particular, the probability of deletions given the AVP was successfully used before (*true*) – or known, indicated, asked – are lower than in the opposite case (*false*). Note that insertion probabilities were not lowered by learning, as insertions often went unnoticed by the user, e.g. when they only affected the dialog flow or when they were ignored in the dialog management.

The *indicated AVPs* can easily be tracked, as they are annotated to the system dialog act representations. E.g., "You can choose German or Italian food!" would be annotated with *choiceValues = {foodtype=German; foodtype=Italian}*.

Table 2. *Error rates in different conditions, where true and false refers to the parameter in each column.*

	success. AVP	*indicate AVP*	*known AVP*	*asked slot*
p(del| true)	.08	.03	.04	.08
p(del| false)	.10	.13	.14	.13
p(ins| true)	.22	.23	.22	.14
p(ins| false)	.18	.17	.16	.23

To obtain a more fine-grained model of memory, we can compute for each AVP at each turn how many exchanges ago it was mentioned (i.e. indicated or successfully used), and how often it has been mentioned so far. This is motivated by the observation that more recent and repeated information can be retrieved from memory more easily. We can then train different confusion matrices for different memory conditions. E.g., a confusion matrix can be learned for only those AVPs which have been successfully used by the user at least three times and not more than five exchanges ago.

4. Results

In order to analyze how these models can support the analysis of experimental data, we use a database collected with the BoRIS restaurant information system [13]. Boris allows users to find restaurants according to the five criteria mentioned above in a mixed-initiative dialog. The database was collected in a Wizard-of-Oz test, where ASR was replaced by one of the experimenters. However, ASR errors were simulated as described in [13]. Also, the NLU capabilities of the system were used to extract meaning from the utterances. Forty Users (29m, 11f; *M* = 29.0y, *SD*=9.7) performed five different tasks. Three dialogs could not be used in the analysis, resulting in 197 dialogs (2001 exchanges) in the entire dataset.

Each dialog was judged on a usability questionnaire [13]. Factor analysis revealed scales related to the overall acceptance of the system (ACC), perceived cognitive effort (COE) and efficiency (EFF; for details, see [15]). In addition, log files are available, listing transcripts of each user and system turn along with speech understanding results and task success annotations.

For the analyses described here, we automatically annotated the system state, the believed system state, and the AVPs indicated by BoRIS to the database. A model of the system was used to determine the system state and prompt annotations at each step, and the described belief model was used to determine the believed system states at each exchange.

4.1. Derivation of parameters

The new annotations made to the corpus allow deriving a number of parameters which may be suitable to describe how the user experienced the interaction, and thus may be correlated with the quality judgments.

First, the edit distance between the believed system state and the user goal can be determined as the number of required, but still empty slots, plus twice the number of wrongly set slots (1 deletion + 1 insertion). As illustrated in Figure 1, this distance can be specified for each exchange in a dialog. Via linear regression, the progress towards the goal can then be quantified as the gradient of the regression line (*GradDist_ALL*). In order to account for the *Recency Effect* (the observation that more recent events often have a higher impact on the judgments [16]), a variant of this measure only considers the progress within the last 3 turns (*GradDist_END*).

Furthermore, the believed system state allows inferring the perceived speech understanding result as the difference between the beliefs in the previous and the current turn. Note that this calculation only considers AVPs belonging to system slots, and not, for example, logical AVPs in affirmation acts. Given the perceived understanding result, a perceived concept error rate (*percCER*) can be computed. Also, the AVPs of which the user knows that she has submitted them successfully (*successful AVPs*) can be determined based on this information.

Alignment to the systems vocabulary may be related to the user's ratings of a dialog as well. In particular, the user may not want to listen to a list of options which are already known. Thus, we measure if the system indicated AVPs already known to the user (*#indiKnownAvps*), and for comparison the number of turns which included any indication of possible values (*#indications*).

4.2. Relation to user judgments

Correlation analysis shows that *GradDist_ALL* is a fair predictor of the ratings on all three quality dimensions, compared to the standard predictors *#Turns* and *Task Success* (Table 3). If we only consider the gradient over the last three exchanges, the correlation with ACC is increased, at the cost of lower correlations on the other aspects. In accordance with intuition, perceived efficiency can be predicted particularly well with these parameters.

Also, it can be seen that the perceived *CER* has a higher correlation with the ratings than the actual *CER*, which includes also errors unnoticed by the user. Strong correlations can also be observed for *#indicate*. In fact, they are higher than those for *#indiKnownAvps*, which is surprising, as indication of concepts helped users to find the right words (Table 2) and should thus have a positive impact in some cases. Figure 2 shows how the correlations change depending on the time passed since the AVP was last mentioned. As expected, the negative impact of indicating AVPs on the judgments decreases if more time passed since its last mentioning, i.e. if a correct wording for the concept was less likely to be remembered. This corresponds to the higher error rates in these cases (Figure 3). Note also that the latter effect becomes more evident if only those AVPs are considered which the system did not ask for in its preceding turn (dashed line).

Table 3. *Correlations of performance and ACC, where * indicates significance and ** high significance*.

Parameter/Rating	ACC	COE	EFF
Task success	.26**	.13	-.03
#Turns	-.34**	-.27**	-.47**
GradDist_ALL	.26**	.28**	.37**
GradDist_END	.33**	.25**	.32**
CER	.00	.05	.15*
percCER	-.21**	-.23**	-.37**
#Indicate	-.30**	-.24**	-.40**
#indiKnownAvps	-.28**	-.18*	-.37**

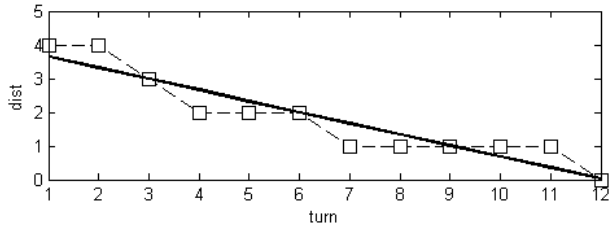


Figure 1: *Distance between believed and desired system state through the 12 exchanges of an example dialog, and regression line through the points.*

Table 4. *Effect of knowing an AVP and being asked for it on error rates for positive compared to negative ratings.*

	all	*ACC > 3*	*ACC <=3*
p(del| known, asked)	.03	.01	.03
p(del| ~known, asked)	.13	.13	.14
p(del| known, ~asked)	.10	.06	.15
p(del| ~known, ~asked)	.14	.13	.14

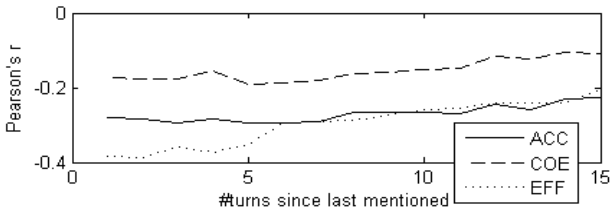


Figure 2. C*orrelation between #indiKnownAvps and judgments, for 1…15 turns since the AVP was last mentioned. All except COE at t=2 are significant.*

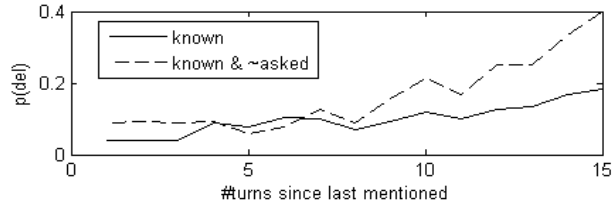


Figure 3. *Error rates for AVPs after 1…15 turns since their last mentioning. The dashed line shows the stronger effect if the AVP was not asked in the preceding prompt.*

Finally, Table 4 shows how successfully using learned vocabulary is related to the user ratings of acceptability. Shown are error rates if the concept was already known (i.e. mentioned) or not, and if the respective slot was asked by the system or not. For dialogs with high acceptance ratings, the effect of knowing an AVP on the error rate is much stronger than for the badly rated dialogs. Note that this effect was only observed for ACC, and not for EFF and COE. Also, the overall error rates were similar. Thus, noticing success in learning to use the system may have improved the users' general attitude towards using it.

5. Discussion & Conclusion

In the previous sections, models for the user's belief about the state of a dialog system, and for the adaptation of speech style, were presented. The models were used to derive parameters which describe the user's experience of the dialog and are thus related to the judgments given by the user on different quality dimensions.

For most of these parameters, significant correlations with the user judgments were obtained, which were in the range of correlations between ratings and *#Turns* or *Task Success*. This also supports the assumption that the proposed models are somewhat realistic estimations of the actual state of mind of the users during the dialogs.

Unfortunately, the results with respect to learned expressions are less clear than those for the belief model. This is mainly due to the fact that users uttered most concepts in direct response to a system question for the same slot, in which case the wording was mostly chosen correctly. A simple explanation for this is that the system's NLU grammar was defined based on the directed prompts to a large part, whereas true mixed-initiative utterances are much more difficult to predict and cover in the grammar.

Nevertheless, some indication was found that the effect of learning vocabulary also impacts the judgments, as a relation between the learning success and overall acceptability of the system was observed.

The presented model can also be used in user simulations to make more reasonable updates of the user state and model temporal aspects of speech understanding errors. However, the main benefit of the method so far lies in post-hoc analysis of data obtained with real users with respect to otherwise unobservable aspects of the user's experience. In the future, more parameters will be derived, including an estimate of subjective task success. In addition, we will analyze how emotions impact the cognitive processes described in this paper.

6. References

[1] Nielsen, J., *Usability Engineering*, San Diego, CA: Academic Press, 1993.

[2] Kieras, D.E., "Model-based evaluation", in: *The Human-Computer Interaction Handbook*, J. Jacko and A. Sears, Eds. Mahwah, NJ: Lawrence Erlbaum Associates, 2003, 1191-1208.

[3] Araki, M. and Doshita, S. "Automatic Evaluation Environment for Spoken Dialogue Systems", in: *Proc. of Workshop on Dialogue Processing in Spoken Language Systems*, 1997, 183-194.

[4] Eckert, W., Levin, E., Pieraccini, R. "User Modeling for Spoken Dialogue System Evaluation", in: *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997.

[5] Schatzmann, J., Thomson, B., Young, S., "Statistical User Simulation with a Hidden Agenda", in: *Proc. 8th SIGdial Workshop*, 2007, 273-282.

[6] Chung, G., "Developing a Flexible Spoken Dialog System Using Simulation", in: *Proc. of ACL 2004*, 2004, 63-70.

[7] López-Cózar, R., Callejas, Z., McTear, M. "Testing the Performance of Spoken Dialogue Systems by Means of an Artificially Simulated User". *Artificial Intelligence Review*, vol. 26, pp. 291-323, 2006.

[8] Pietquin, O. *A framework for unsupervised learning of dialogue strategies*, Ph.D. thesis, Faculty of Engineering, Mons (TCTS Lab), Belgium, 2004.

[9] Engelbrecht, K.-P. *Estimating Spoken Dialog System Quality with User Models*, dissertation, Technische Universität Berlin, 2012.

[10] Ai, H., Weng, F. "User Simulation as Testing for Spoken Dialog Systems", in: *Proc. of SIGdial 2008*, 2008, 164-171.

[11] Bernsen, N. O., Dybkjær, H., and Dybkjær, L. *Designing Interactive Speech Systems: From First Ideas to User Testing*. Berlin: Springer, 1998.

[12] Walker, M., Litman, D., Kamm, C., Abella, A.,"PARADISE: A Framework for Evaluating Spoken Dialogue Agents", in: *Proc. of ACL/EACL*, 1997, 271–280.

[13] Möller, S. *Quality of Telephone-based Spoken Dialog Systems*. New York: Springer, 2005.

[14] S. Young, M. Gasic, S. Keizer, …, K. Yu (2010). "The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management." Computer Speech and Language 24(2), pp. 395-429, 2010.

[15] Möller, S., Engelbrecht, K.-P., Schleicher, R., "Predicting the Quality and Usability of Spoken Dialogue Services," *Speech Communication*, vol. 50, pp. 730-744, 2008.

[16] Hassenzahl, M., Sandweg, N.. "From Mental Effort to Perceived Usability: Transforming Experiences into Summary Assessments," in: *Proc. of CHI 2004*, 2004, 1283–1286.