# Cluster-based User Simulations for Learning Dialogue Strategies

*Verena Rieser* and Oliver Lemon†*

*Department of Computational Linguistics, Saarland University, Saarbrücken, Germany
†School of Informatics, University of Edinburgh, Edinburgh, UK
vrieser@coli.uni-sb.de, olemon@inf.ed.ac.uk

## Abstract

Good dialogue strategies in spoken dialogue systems help to ensure and maintain mutual understanding and thus play a crucial role in robust conversational interaction. We focus on clarification strategies and build user simulations which are critical for reinforcement learning, which is a cheap and principled way to automatically optimise dialogue management. In this paper we present a novel cluster-based technique for building user simulations which show *varying*, but *complete* and *consistent* behaviour with respect to real users. We use this technique to build user simulations and we also introduce the SUPER evaluation metric which allows us to evaluate user simulations with respect to these desiderata. We show that the cluster-based user simulation technique performs significantly better (at $P < 0.01$) than decisions made using either the one most likely action or a random baseline. The cluster-based user simulations reduce the average error of these other models by 53% and 34% respectively.

**Index Terms**: spoken dialogue, user simulation, evaluation metrics, reinforcement learning, dialogue strategies

## 1. Introduction

Good dialogue strategies in spoken dialogue systems help to ensure and maintain mutual understanding and thus play a crucial role in robust conversational interaction. In this work we focus on clarification strategies. The overall goal is to learn a clarification strategy which is adaptive, portable to other domains, robust and natural. Therefore we bootstrap a clarification strategy from data gathered in a Wizard-of-Oz study and optimise this strategy using reinforcement learning, as described in [1, 2]. The reinforcement learning approach to dialogue management is a cheap and principled way to optimise dialogue systems which act in a robust but flexible way. Exploratory trial-and-error learning with real users might be a time-consuming and sadistic procedure, so instead of having real users interacting with the system we apply user simulations for learning. The quality of the learnt strategy depends on the quality of the user simulation in the sense that the simulated user provides responses to the system actions which allow the system to explore the state space. In this paper we present new techniques for building and evaluating user simulations. In previous work user simulations tried to resemble real users by using n-gram models or supervised learning techniques (see for example [3]). As argued by [4], the desiderata on user simulations are *naturalness* and *variety*.

In this paper we show that cluster-based user simulations are significantly better than methods based on either most likely single actions or a random baseline with respect to those goals. Furthermore we show that our user simulations are *complete*, and *consistent*, while showing *variance* in their behaviour. Having some variance is essential for reinforcement learning to explore a large enough state space to learn a strategy which is robust to less frequent user actions.

We proceed as follows. In section 2 we summarise the data collection in a Wizard-of-Oz study. We describe the annotation scheme and report reliability. In section 3 we explain cluster-based user simulation techniques, and describe three different models (based on different feature spaces). In section 4 we present the SUPER evaluation scheme, which calculates a **S**imulated **U**ser **P**ragmatic **E**rror **R**ate. In section 5 we show that cluster-based user models are significantly better than models which are based on single most likely actions and significantly better than random. Finally, we explain how we plan to test the domain-independce of our user simulation in order to learn a clarification strategy which is portable to other domains of information seeking dialogue.

## 2. The Data

### 2.1. The Corpus

The corpus we are using for bootstrapping clarification strategies was collected in a multimodal Wizard-of-Oz study of German task-oriented dialogues for an in-car music player application, [5]. In this study six wizards played the role of an intelligent interface to an MP3 player and were given access to a database of information. 24 subjects were given a set of predefined tasks and were told to accomplish them by using an MP3 player with a multimodal interface. This environment introduced uncertainties on several levels, for example multiple matches in the database, lexical ambiguities, and errors on the acoustic level, as described in [5]. The corpus gathered with this setup comprises 70 dialogues, 1772 turns and 17076 words. Example 1 shows a typical clarification sub-dialogue (translated from German).

### 2.2. Annotation Scheme

The data is annotated with the following annotation scheme for for clarification requests (CRs) which has been shown to be applicable for several different domains of dialogue [6]. In this scheme a clarification object is a triple of three related utterances; one utterance being the CR itself, the antecedent (i.e. the problematic user utterance which caused the CR) and the reply to that CR. For each of these three utterances we are annotating additional attributes as shown in figure 1. For the CR itself we annotate the problem `source` and degree of uncertainty (`severity`) as indicated by the speaker. The problem source of the clarification request describes the type of understanding problem which caused the need to clarify. Its attributes map to the level of understanding as defined by [7]. The problem severity describes which type of feedback the CR-initiator requests from the other dialogue participant, i.e. asking for `confirmation` or for `elaboration/repetition`. For the antecedent we are interested in its speech act type and its

$$\left[\begin{array}{l} \text{ANTECEDENT: } \left[\begin{array}{l} \texttt{SA-action:} \left\{\texttt{command} \mid \texttt{provide-info} \mid \texttt{request-info} \mid \texttt{confirm}\right\} \\ \texttt{SA-arguments:} \left\{\texttt{select} \mid \texttt{search} \mid \texttt{show} \mid \texttt{add} \mid \texttt{play} \mid \texttt{delete} \mid \texttt{make} \mid \texttt{list} \mid \texttt{NULL}\right\} \end{array}\right] \\ \text{CR: } \left[\begin{array}{l} \texttt{source} \left\{\texttt{contact} \mid \texttt{acoustic} \mid \texttt{goal/reference} \mid \texttt{intention}\right\} \\ \texttt{severity:} \left\{\texttt{confirmation} \mid \texttt{repetition/elaboration}\right\} \end{array}\right] \\ \text{REPLY: } \left[\texttt{reply-type:} \left\{\texttt{add} \mid \texttt{repeat} \mid \texttt{y/n} \mid \texttt{change}\right\}\right] \end{array}\right]$$

Figure 1: The annotation scheme

arguments as shown in example 1. The reply is classified according to its information gain and the complexity of the underlying language model. These attributes reflect that a good clarification strategy for spoken dialogue systems should elicit responses which maximise the information gain while minimising recognition errors. These desiderata are reflected in the values of the reply type, which are adding information (`add`), repeating an utterance (`repeat`), a y/n answer (`y/n`), or the user changes topic (`change`). The following example shows how one clarification sub-dialogue got annotated.

(1) **[User:** ] Please show me the playlist.
**Antecedent:** `SA-action= command`
`SA-argument= show`

**[Wizard:** ] Which playlist do you mean?
**CR:** `source= reference`,
`severity= repetition/elaboration`

**[User:** ] Beatles.
**Reply:** `reply-type= add`

### 2.3. Reliability

The whole annotation was performed twice, by an expert and by a naïve annotator. For evaluating the reliability of the manual annotations we used the $\kappa$ coefficient. For identifying CRs we chose a cascaded approach as introduced by [8], to assure maximal reliability for this task with $\kappa > 0.8$. For annotating further features we only used the cases which both annotators identified as being clarification requests, resulting in 155 annotated CRs. The reliability of all the other features listed above was in-between the accepted boundaries ($0.67 < \kappa < 0.8$).

## 3. Cluster-based User Simulations

Based on these annotations we built user simulations that generate any value of `reply-type` based on CR and antecedent features. Rather than building an accurate model of 'average' user behaviour in the data as most user simulations based on n-gram and supervised learning do, we want our user simulations to generate *complete* and *consistent* sets of all observed possible actions according to their observed frequency in each context. That is, the user simulations should be able to produce any kind of observed user behaviour in a context (as opposed to only the overall 'average' behaviour), but should not generate impossible actions. Ultimately, this will allow us to learn a strategy which is robust to less frequent user behaviour.

To construct a user simulation that behaves in such a way, we apply a cluster-based approach. Clustering groups together feature vectors based on their similarity. We first build clusters on the features of CR and antecedent using the probability based Expectation-Maximization algorithm for cluster assignment. Then we inspect the probability distribution of `reply-type` for each cluster. That is, we abstract away from a specific combination of context features, and cluster together situations which are similar w.r.t. feature vectors, and investigate the range of possible behaviours in those situations.

We generate these clusters based on different feature sets, resulting in three different user simulation models. For the first model we only take features from the CR, representing a minimal local context, which leads to two clusters. For the two other simulations we also take features from the antecedent leading to four and two clusters respectively (see table 1).

| model | features | cluster size |
|---|---|---|
| sim1 | `source, severity` | (134/21) |
| sim2 | `source, severity, SA-action` | (21/8/66/60) |
| sim3 | `source, severity, SA-argument` | (128/27) |

Table 1: Cluster-based user simulations

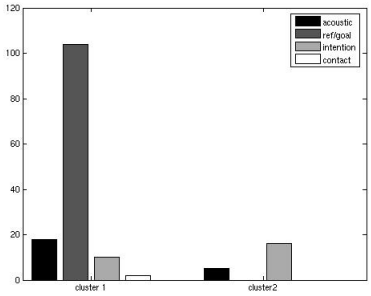


Figure 2: Distribution of `source` between two clusters for `sim1`.

This approach has several advantages over standard user simulations such as n-grams or supervised learning techniques. By applying unsupervised learning instead of supervised, we avoid the bias-variance problem in the model selection process, but we model all possible observations while still being able to handle unseen events by assigning them to clusters. [1] This will allow us to learn clarification strategies which are more robust and flexible in

[1] Note that n-gram models sometimes apply smoothing techniques, i.e. they assign a low probability to unseen events. That is, smoothing produces in-constitent user behaviour. Supervised learning techniques, for example decision trees, use pruning to reduce the bias. Note that this technique causes a user simulation to be incomplete.

**Consistency:**
`if(P0(action) =0 and P1(action)>0)`: I = (-1)
**Completeness:**
`if(P0(action)>0 and P1(action)=0)`: D = (-1)
**Desired variation:**
`if(|P0(action)-P1(action)|<`$\epsilon$): V = (+1)
**Tolerated variation:**
`if(`$\epsilon <$`|P0(action-P1(action)|<`$\delta$): V = (0)
**Penalised variation:**
`if(`$\delta <=$`|P0(action)-P1(action)|)`: V = (-|P0(action)-P1(action)|)

Figure 3: Pseudo-code for SUPER evaluation rules

.

less frequent situations using only a limited amount of data, i.e. we are able to explore more fully the space of all possible states and actions. [2] Furthermore, by abstracting away from specific feature combinations we simplify the model. Our user simulations only need to decide whether they are in one specific cluster. When taking this decision, the cluster-based approach also allows us to incorporate uncertainty about the features, since one cluster can contain many different values for the same feature, see figure 2. This is especially important for handling interpretation uncertainty in a working system, w.r.t. the antecedent and CR features.

## 4. SUPER User Simulation Evaluation

In this section we evaluate the different user simulation with respect to a proposed **S**imulated **U**ser **P**ragmatic **E**rror **R**ate (SUPER). For evaluating user simulations we don't want to measure how well we are to able resemble behaviour of an 'average' user, i.e. we don't want to measure accuracy, but as argued by [4] the evaluation must cover aspects of *naturalness* and *variety* of user behaviour. In [4] three different ways to evaluate user simulations are suggested: (1) high-level dialogue features, such as dialogue length; (2) dialogue style, such as the frequency of the different speech acts; and (3) success rate and efficiency of the dialogues. However, none of these features capture the fact that we want a user simulation that shows *varying* behaviour, but also is *complete* and *consistent*. These principles are captured by the SUPER score rule set shown in figure 3 where $P0$ is the observed `reply-type` label probability for one feature combination (i.e. a context) and $P1$ the probability assigned by one cluster (i.e. a simulated user). Note that the SUPER score also captures a number of the principles of WER for speech recognition, but with variation allowed (c.f. BLEU).

- **Consistency:** The user simulation should not do things that real users wouldn't do in this context. i.e. no insertions (I).
- **Completeness:** The user simulation should produce every possible action of real users, i.e. no deletions (D).
- **Variation:** The user simulation should behave *like* the real users but not duplicating the average behaviour with 100% accuracy. Therefore we define a lower boundary $\epsilon$ which reflects the desired variation (V), and an upper boundary $\delta$ which reflects undesired variation, i.e. the simulation behaves in a more flexible way. Note that $\epsilon$ and $\delta$ values can be used specify needs of the application domain.

[2] Note that even for the few features chosen for our small data set we end up learning a strategy for $2^4, (2^4)^4, (2^4)^9$ possible state space combinations.

For assigning negative scores we treat variation less harshly than penalising insertions and deletions, to account for these cases having a different severity for learning a dialogue strategy. For evaluation we compare the probability for one `reply-type` label assigned by the cluster-based user simulation against the probability of that `reply-type` label for a specific context observed in the original data. For example for the user simulation `sim1` we assign every instance the `reply-type` probability for its `severity` and `source` feature values ($P0$). Then we compare the observed probability $P0$ against the expected probability from the cluster the instance is assigned to ($P1$), and apply the evaluation rules listed in figure 3. We do this for every instance in our corpus, and then sum and normalise over actions and contexts, resulting in a SUPER score. The SUPER score evaluation technique can be expressed as in equation 2, where $n$ is the number of actions in a context (e.g. `reply-type`), $m$ is the number of contexts, and where V, I, and D are defined as above for each context $C_m$. The SUPER score is defined over $[-1, +1]$.

$$\text{SUPER} = \frac{1}{m}\sum_{k=1}^{m}\frac{V+I+D}{n} \quad (2)$$

## 5. Evaluation Results

For evaluating our user simulations we compare the three cluster-based models against a random baseline and a majority baseline which predicts the most frequent `reply-type` for each feature combination (see figure 2). The majority baseline corresponds to a more basic n-gram user simulation technique which only picks the most frequent single action in a single context (e.g. [9]). For our example the feature combination $f_{r,r}$= (`source= reference, severity= repetition`) and $f_{a,c}$ = (`source= acoustic, severity= confirm`) the probability distributions are displayed in figure 2. For $f_{a,c}$ the observed distribution $P0$ of `reply-type` is only binary, i.e. in the real data this feature combination was only observed together with these two types. Whereas for $f_{r,r}$ the full range of reply types are observed. $P0$ with zero counts for one `reply-type` may result in a negative score for insertions (i.e. condition ($P0 = 0$ `and` $P1 > 0$) matches for $f_{a,c}$). Note that these non-occurences can also be due to data sparsity, e.g. for $f_{a,c}$ we have only 5 counts in our data, whereas the combination $f_{r,r}$ was observed 93 times. Thus, for evaluating the user simulations `sim2` and `sim3`, which are based on larger contexts, feature combinations which were seen less than 4 times are pruned away to avoid artefacts intro-

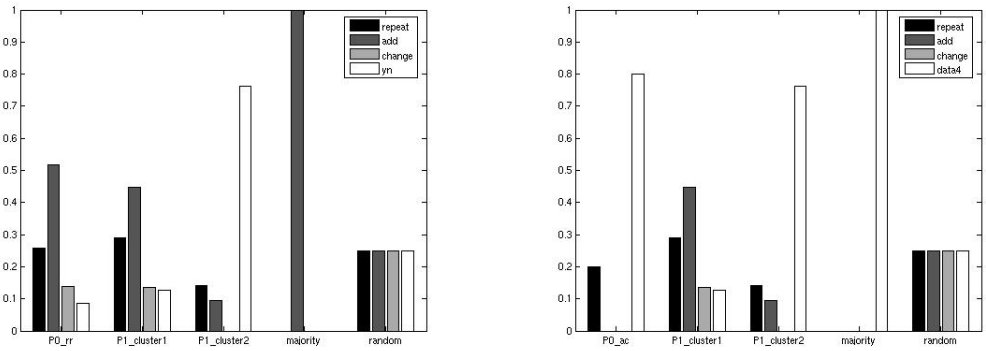


Table 2: Probability distribution of `reply-type` for $f_{r,r}$ (left) and $f_{a,c}$ (right).

duced by sparse data. [3] The results are shown in table 3, where $\epsilon = 0.1$ and $\delta = 0.4$ which reflects our goal to have a reasonable amount of variation. A paired samples t-test on the individual scores showed that for all simulation models, the cluster-based simulation method is significantly better than the majority baseline (with corrected $\alpha$ value of $P < 0.01$), and thus significantly better than more basic user simulations that work on the basis of selecting the single most likely action.[4] The best performing cluster-based model is `sim3` which is significantly better than the random baseline. There is no significant difference between the mean SUPER scores of the cluster-based techniques.

The cluster-based simulations reduce the average error of the majority class simulations by 53%, and the average error reduction with respect to the random simulations is 34%. This score is calculated via distance from the perfect SUPER score of 1.

| model | cluster | majority | random |
|---|---|---|---|
| `sim1` | 0.48 | -0.028 | 0.14 |
| `sim2` | 0.43 | -0.026 | 0.31 |
| `sim3` | 0.54 | -0.24 | 0.22 |

Table 3: SUPER scores for cluster-based user simulations

## 6. Summary and Future Work

In this paper we present a cluster-based technique for building user simulations which show varying, but complete and consistent behaviour with respect to real users. We use this technique to build user simulations which can be used to learn clarification strategies which are robust and portable to different domains of information-seeking dialogue. We first described the data collection and in a Wizard-of-Oz study, the annotation scheme for clarification subdialogues, and we reported on reliability. We argued that for learning a portable clarification strategy which is robust to new and less frequent user actions we need user simulations that cover all possible actions of real users (i.e. is *complete*), but should not do things a real user would not do in this context (i.e. is *consistent*), and also should include some *variation* to explore the state space more fully. We explained how to build user simulations using a cluster-based technique, where we cluster similar dialogue contexts and assign a probability distribution for possible user replies in those contexts. We built three different user simulations based on different context definitions. Then we introduced the SUPER evaluation metric which allows us to evaluate user simulations. We show that the cluster-based technique is significantly better than decisions made using the one most likely action. The best performing cluster-based model (`sim3`) is defined over the feature space `source`, `severity` and `SA-action`, and is significantly better ($P < 0.01$) than the random baseline.

In future work we will build a cluster-based user simulation for the flight information domain to test the domain-independence of our user simulations in order to be able to learn clarification strategies which are portable to other domains.

[3] Note, that pruning is only applied to the unclustered data for evaluation purposes. The cluster-based user models do not apply any pruning as noted in section 3.

[4] Note that this also implies that simulations based on probability distributions (e.g. [3]) perform better towards our desiderata.

## 7. References

[1] V. Rieser, I. Kruijff-Korbayová, and O. Lemon, "A corpus collection and annotation framework for learning multimodal clarification strategies," in *Proceedings of SIGdial6*, 2005.

[2] V. Rieser and O. Lemon, "Using machine learning to explore human multimodal clarification strategies," in *Proceedings of the 44rd ACL*, 2006.

[3] K. Georgila, J. Henderson, and O. Lemon, "Learning user simulations for information state update dialogue systems," in *Proceedings of INTERSPEECH*, 2005.

[4] J. Schatzmann, K. Georgila, and S. Young, "Quantitative evaluation of user simulation techniques for spoken dialogue systems," in *Proceedings of SIGdial6*, 2005.

[5] I. Kruijff-Korbayová, T. Becker, N. Blaylock, C. Gerstenberger, M. Kaißer, P. Poller, J. Schehl, and V. Rieser, "An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system," in *Proceedings of ENLG*, 2005.

[6] V. Rieser and J.D. Moore, "Implications for Generating Clarification Requests in Task-oriented Dialogues," in *Proceedings of the 43rd ACL*, 2005.

[7] H. Clark, *Using Language*, Cambridge University Press, 1996.

[8] J. Carletta, A. Isard, S. Isard, J. Kowtko, G.h Doherty-Sneddon, and A. Anderson, "The reliability of a dialogue structure coding scheme," *Computational Linguistics*, vol. 1, no. 23, pp. 13–31, 1997.

[9] Roberto Pieraccini Esther Levin and Wieland Eckert, "A stochastic model of human machine interaction for learning dialog strategies.," in *IEEE Transactions on Speech and Audio Processing*, 2000.