

# CLASSiC

---

## D3.3: Simulated users for training NLG (TownInfo/Self-Help)

---

Verena Rieser, Srinivasan Janarthanam, Oliver Lemon

Distribution: Public

---

CLASSiC

Computational Learning in Adaptive Systems for Spoken Conversation  
216594 Deliverable 3.3

July 2010



Project funded by the European Community  
under the Seventh Framework Programme for  
Research and Technological Development



*The deliverable identification sheet is to be found on the reverse of this page.*

<b>Project ref. no.</b>	216594
<b>Project acronym</b>	CLASSiC
<b>Project full title</b>	Computational Learning in Adaptive Systems for Spoken Conversation
<b>Instrument</b>	STREP
<b>Thematic Priority</b>	Cognitive Systems, Interaction, and Robotics
<b>Start date / duration</b>	01 March 2008 / 36 Months

<b>Security</b>	Public
<b>Contractual date of delivery</b>	M24 = February 2010
<b>Actual date of delivery</b>	July 2010
<b>Deliverable number</b>	3.3
<b>Deliverable title</b>	D3.3: Simulated users for training NLG (TownInfo/Self-Help)
<b>Type</b>	Prototype
<b>Status &amp; version</b>	Final 1.1
<b>Number of pages</b>	17 (excluding front matter)
<b>Contributing WP</b>	3
<b>WP/Task responsible</b>	WP3, leader UEDIN
<b>Other contributors</b>	Xingkun Liu, Helen Hastie
<b>Author(s)</b>	UEDIN: Verena Rieser, Srinivasan Janarthanam, HW: Oliver Lemon
<b>EC Project Officer</b>	Philippe Gelin
<b>Keywords</b>	Simulated users, Natural Language Generation, evaluation

The partners in CLASSiC are:

<b>Heriot-Watt University</b>	HWU
<b>University of Cambridge</b>	UCAM
<b>University of Geneva</b>	GENE
<b>Ecole Supérieure d'Electricité</b>	SUPELEC
<b>France Telecom/ Orange Labs</b>	FT
<b>University of Edinburgh HCRC</b>	EDIN

For copies of reports, updates on project activities and other CLASSiC-related information, contact:

The CLASSiC Project Co-ordinator:  
 Dr. Oliver Lemon  
 School of Mathematical and Computer Sciences (MACS)  
 Heriot-Watt University  
 Edinburgh  
 EH14 4AS  
 United Kingdom  
 O.Lemon@hw.ac.uk  
 Phone +44 (131) 451 3782 - Fax +44 (0)131 451 3327

Copies of reports and other material can also be accessed via the project's administration homepage,  
<http://www.classic-project.org>

No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.

# Contents

<b>Executive Summary</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
<b>2 N-gram user simulations models for the TownInfo domain</b>	<b>2</b>
2.1 Purpose of the user simulation . . . . .	2
2.2 Data . . . . .	3
2.3 Method . . . . .	6
2.4 Evaluation . . . . .	7
2.5 Brief description of the code . . . . .	8
<b>3 A Two-tier User Simulation Model for the Self-Help domain</b>	<b>9</b>
3.1 Purpose of the user simulation . . . . .	9
3.2 Data . . . . .	9
3.3 Method . . . . .	9
3.3.1 Advanced n-gram model . . . . .	10
3.3.2 A Two-tier model . . . . .	10
3.3.3 Baseline Bigram model . . . . .	11
3.3.4 Trigram model . . . . .	11
3.3.5 Equal Probability model baseline . . . . .	11
3.3.6 Smoothing . . . . .	11
3.4 Evaluation . . . . .	12
3.5 Brief description of the code . . . . .	13
<b>4 Comparison</b>	<b>13</b>
<b>5 Conclusion</b>	<b>13</b>

## Executive summary

In the CLASSiC project, we are the first to develop data-driven simulated users which are sensitive to NLG choices. This report describes two prototypes of simulated users for training statistical NLG: n-gram user simulations to optimise Information Presentation strategies in the TownInfo domain; and two-tier user simulations to optimise Referring Expression Generation in the Self-Help domain. We describe the methods used to build the user simulations from small amounts of Wizard-Of-Oz data. In both domains the models use back-off and discounting techniques to overcome data sparsity. We evaluate the user simulations using the Kullback-Leibler (KL) divergence to measure dialogue similarity. We conclude that the best models in both domains produce dialogue actions which are close to the ones observed in the original data, given low KL scores in both domains ( $KL_{SelfHelp} = 0.011$  and  $KL_{TownInfo} = 0.018$ ).

We also briefly describe the implementation of the user simulations, which is currently used to optimise NLG strategies using Reinforcement Learning.

(Some aspects of this work were published at SigDial 2009 [Janarthanam and Lemon, 2009a] )

# 1 Introduction

We present two prototypes of user simulations used to optimise Natural Language Generation (NLG) in Spoken Dialogue Systems (SDS).

A user simulation is a predictive model of real user behaviour and commonly used for automatic dialogue strategy development and testing. See [Schatzmann et al., 2006] for a comprehensive survey. Several techniques for building such simulations have been proposed, e.g. [Eckert et al., 1997], [Georgila et al., 2006a], [Rieser and Lemon, 2006], [Ai and Litman, 2007], [Schatzmann et al., 2007]. However, those models are only sensitive to dialogue management decision and interact with the learner using high-level dialogue acts.

In the CLASSiC project, we are the first to develop data-driven simulated users which are sensitive to NLG choices. In the following report, we describe two prototypes of NLG user simulation. In Section 2, we present a user simulation model for the TownInfo domain which is responsive to different Information Presentation strategies, as described in [Rieser and Lemon, 2009], [Rieser and Lemon, 2010]. In Section 3, we present a two-tier user simulation for the Self-Help domain which is sensitive to referring expressions generated by the system [Janarthanam and Lemon, 2009a], [Janarthanam and Lemon, 2010]. In Section 4, we compare the models from both domains.

We build these user simulations from small amounts of Wizard-of-Oz (WoZ) data, as described in CLASSiC Deliverable 6.1.1., see [Boidin et al., 2009]. We introduce methods to handle data-sparsity. We show that our models outperform other baseline models, using the Kullback-Leibler divergence as a dialogue similarity measure.

## 2 N-gram user simulations models for the TownInfo domain

### 2.1 Purpose of the user simulation

In this Section, we present a user simulation prototype which is used to optimise different Information Presentation decisions in the TownInfo domain, as described in [Rieser and Lemon, 2009, Rieser and Lemon, 2010].

The user simulation is sensitive to different NLG choices. In particular, it is reactive to the generated Information Presentation (IP) strategy, the number of attributes (1-10), and the previous user act. We investigate the following seven IP strategies:

1. RECOMMEND
2. COMPARE
3. SUMMARY
4. COMPARE+RECOMMEND
5. SUMMARY+RECOMMEND
6. SUMMARY+COMPARE

## 7. SUMMARY+COMPARE+RECOMMEND

In response to these IP strategies, the user simulation generates a distribution of likelihoods over the following seven user actions:

1. **select**: the user chooses one of the presented items, e.g. *“Yes, I take this one.”* This reply type indicates that the information presentation was sufficient for the user to make a choice.
2. **requestMoreInfo**: the user asks for more information, e.g. *“Can you recommend me one?”*, *“What is the price range of the last item?”* This reply type indicates that the system failed to present the information the user was looking for.
3. **addInfo**: the user provides more attributes, e.g. *“I am searching for something cheap.”* This reply type indicates that the user has more specific requests, which s/he is able to specify after being presented with the current information.
4. **askRepeat**: the user asks the system to repeat the same message again, e.g. *“Can you repeat?”* This reply type indicates that the utterance was either too long for the user to remember, or the TTS quality was not good enough, or both.
5. **silence**: the user is not saying anything. In this case it is up to the system to take initiative.
6. **hangup**: the user slams down the phone.
7. **other**

## 2.2 Data

We now describe the distribution of user acts in the data.

We build this user simulation from WoZ data collected within the CLASSiC project. This corpus contains 213 dialogues, 18 subjects and 468 NLG actions. A detailed description can be found in Deliverable 6.1.1. [Boidin et al., 2009] and in [Liu et al., 2009].

Figure 1 shows the frequency of user acts in the data set. Most of the time users select an item, request more information, or add more information.

For building user simulations, we are especially interested in user acts in a certain context, i.e. after a system action. Figure 2 shows the percentage of each user act after an IP strategy.

It is interesting to note that the users only ever ask the system to repeat after hearing a summary (100% **askRepeat**, see Figure 2). The only time users stay silent is after COMPARE (100% **silence**, see Figure 2). The case where the user hangs up the phone happens five times in the whole corpus. Three of those cases are after IP strategies.<sup>1</sup> A closer analysis of the data shows that these three cases are due to the noise, i.e. the system repeatedly presenting the wrong (noisy) information to the user.

---

<sup>1</sup>The other two are after no matches were found and a system error.

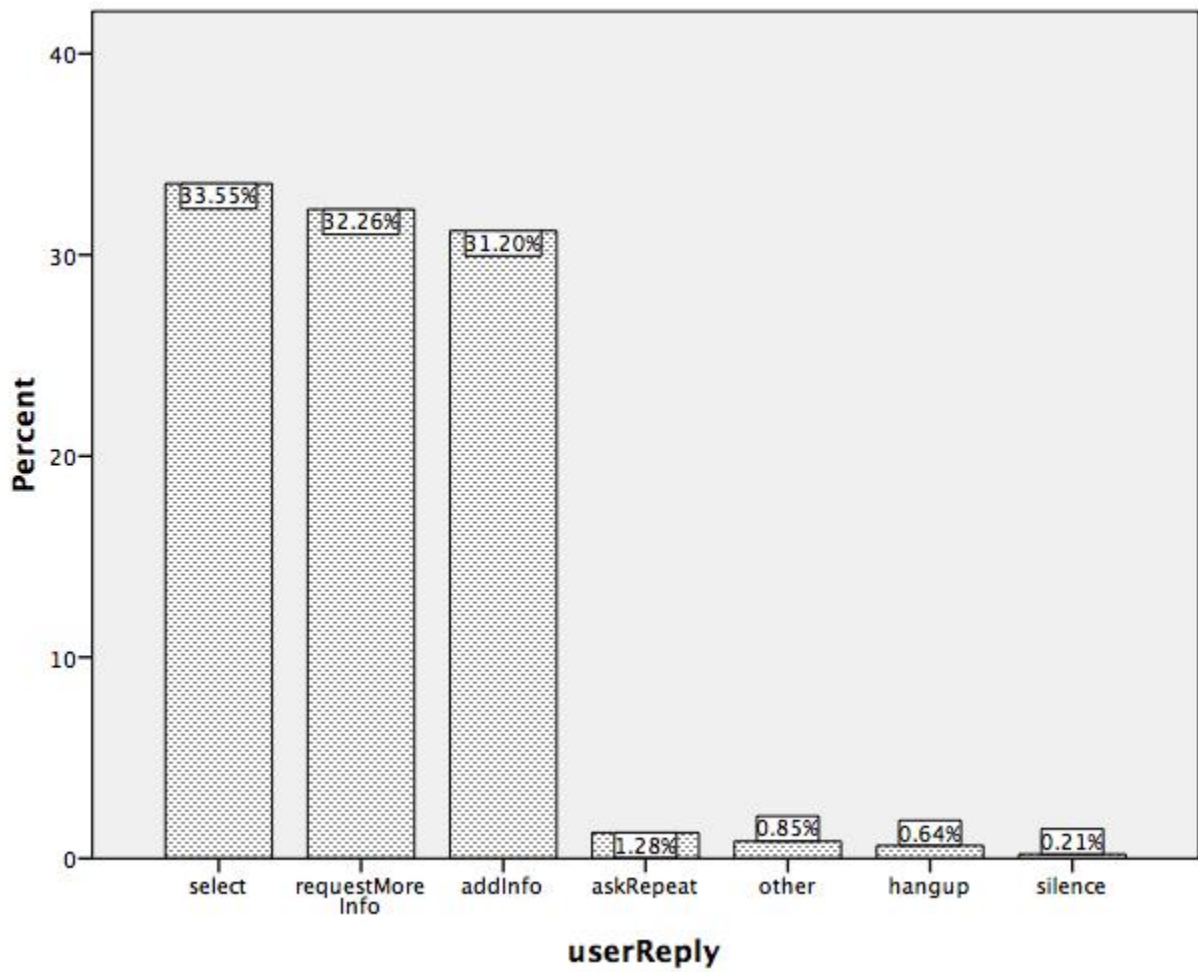


Figure 1: User act frequencies

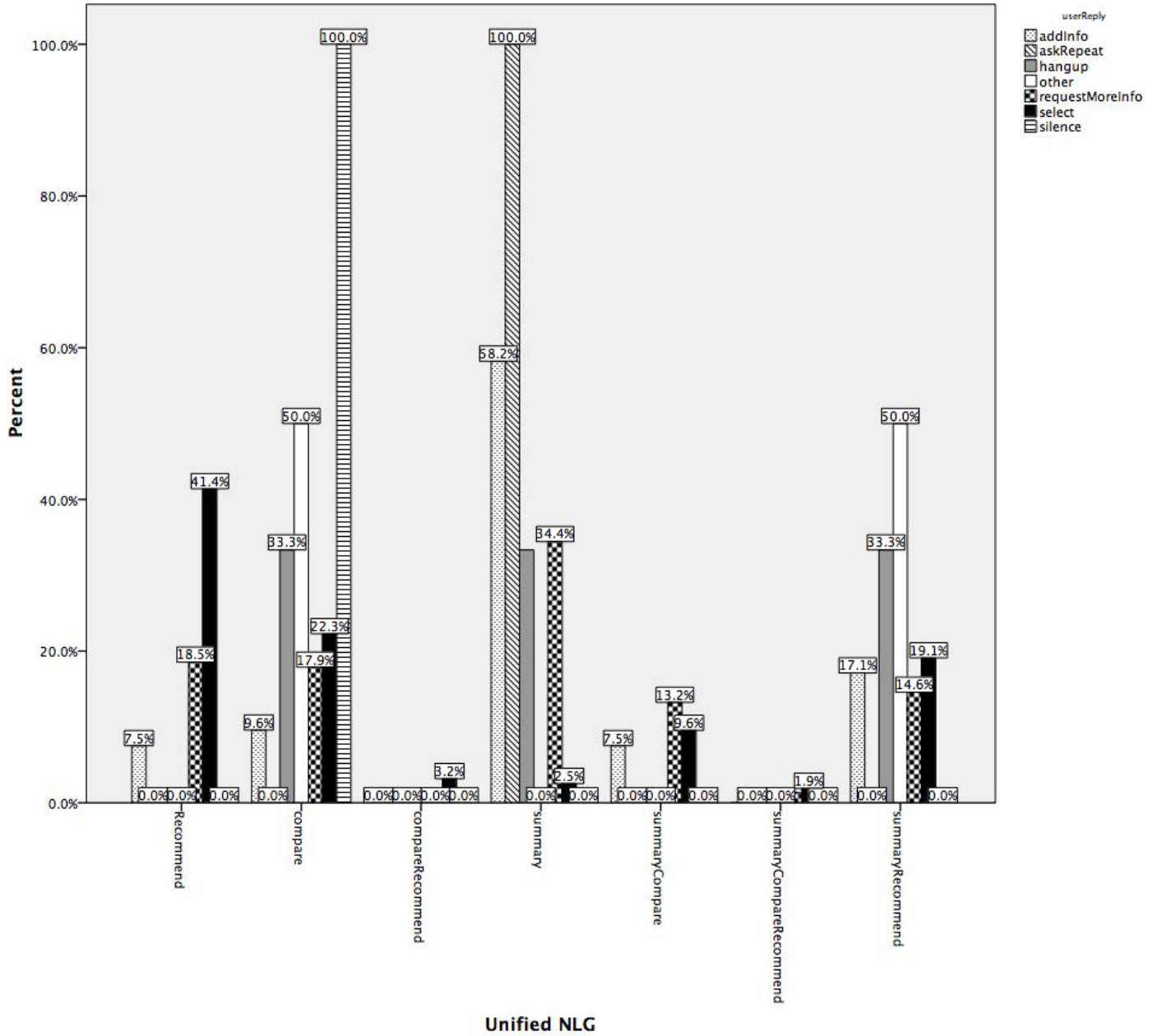


Figure 2: User reply frequencies after an IP strategy

Furthermore, note that some user act types have zero frequencies for almost every IP act. The same is true for number of attributes.<sup>2</sup> However, user simulations for Reinforcement Learning should not include zero probabilities, as they should allow the learner to also find strategies, which are not in the data. In particular, user simulations for automatic strategy *training* should cover the whole variety of possible user action for each state in order to produce robust strategies, especially when learning from small data sets. [Ai et al., 2007], for example, show that random models outperform than more accurate ones if the latter fail to provide enough coverage due to missing data. We, therefore, experiment with techniques to overcome the data sparsity problem for user simulations.

## 2.3 Method

In order to account for the sequential nature of dialogue, we model user simulations as n-gram models of system and user acts. Bi-gram (or more general n-gram) models for user simulations are first introduced by [Eckert et al., 1997, Eckert et al., 1998]. An n-gram based user simulation predicts the estimated user action  $\hat{a}_{u,t}$  at time  $t$  that is most probable given the dialogue history of system and user actions, given the Markov assumption (see Equation 1 where  $a_{s,t}$  denotes the system action at time  $t$ ).

$$\hat{a}_{u,t} = \operatorname{argmax}_{a_{u,t}} P(a_{u,t} | a_{s,t-1}, a_{u,t-1}, \dots, a_{u,0}, a_{s,0}) \quad (1)$$

In order to account for data sparsity, we apply *discounting* (“smoothing”) techniques, using the CMU Statistical Language Modelling (SLU) toolkit [Clarkson and Rosenfeld, 1997]. Discounting is the process of replacing the original counts with modified counts so as to redistribute the probability mass from the more commonly observed events to the less frequent and unseen events. We investigate the following discounting techniques:

- Witten-Bell, [Witten and Bell, 1991];
- Good turing, [Katz, 1987];
- Absolute discounting, [Manning and Schuetze, 1999];
- Linear discounting, [Ney et al., 1994].

Note that the CMU SLU toolkit also applies automatic *back-off* from n-gram model to a more general (n-1) model if the current model is unable to make a reliable prediction due to missing data.

We build two models of different complexity, which predict the probability of a user act ( $a_{u,t}$ ) at time  $t$ , given different contexts/ histories of events:

---

<sup>2</sup>A similar graph was drawn for number of attributes, which shows zero frequencies of user acts for all of the cases. We decided not include the graph in the report since it is difficult to interpret.

- **Bi-gram** model for training the system’s IP strategy decision:  $P(a_{u,t}|IP_{s,t})$ ;
- **Tri-gram** model for training strategy IP strategy decision with integrated attribute selection:  $P(a_{u,t}|IP_{s,t}, attributes_{a,t})$ .

## 2.4 Evaluation

Several metrics have been discussed to evaluate user simulations , e.g. [Schatzmann et al., 2005], [Georgila et al., 2006b], [Scheffler and Young, 2001], [Ai and Litman, 2006], [Williams, 2007]. We measure dialogue similarity based on the Kullback-Leibler (KL) divergence <sup>3</sup>, as also used by e.g [Cuayáhuitl et al., 2005, Jung et al., 2009], which is defined as follows:

$$DS(P||Q) = \frac{1}{N} \sum_{i=1}^N \frac{D_{KL}(P||Q)+D_{KL}(Q||P)}{2}$$

$$D_{KL}(P||Q) = \sum_{i=1}^M p_i * \log\left(\frac{p_i}{q_i}\right)$$

The metric measures the divergence between distributions  $P$  and  $Q$  in  $N$  different contexts with  $M$  responses per context. Ideally, the dialogue similarity between two similar distributions is close to zero.

We compare the raw probabilities as observed in the data with the probabilities generated by our n-gram models using different discounting techniques for each context, see table 1. All the models have a small divergence from the original data (especially for the bi-gram user simulation) suggesting that they are reasonable simulations for training and testing NLG policies.

method	bi-gram	tri-gram
WittenBell	0.086	0.512
GoodTuring	0.086	<b>0.163</b>
absolute	0.091	0.246
linear	<b>0.011</b>	0.276

Table 1: Kullback-Leibler divergence between probabilities observed in original data and probabilities as generated by the bi-gram model  $P(a_{u,t}|IP_{s,t})$  using different discounting techniques.

The absolute discounting method for the bi-gram model is most dissimilar to the data, as is the WittenBell discounting method for the tri-gram model, i.e. the models using these discounting methods have the highest KL score. The best performing methods (i.e. most similar to the original data), are linear discounting for the bi-gram model and GoodTuring for the tri-gram. We use the most similar user models for system training, and the most dissimilar user models for testing NLG policies, in order to test whether the learned policies are robust and adaptive to unseen dialogue contexts.

<sup>3</sup>Also known as information divergence, information gain, or relative entropy

We also report on Perplexity, which is commonly used to evaluate language model n-grams and was also used to evaluate user simulation [Georgila et al., 2006b].

- The bi-gram model  $P(a_{u,t}|IP_{s,t})$  has a perplexity of 44.29.
- The tri-gram model  $P(a_{u,t}|IP_{s,t}, attributes_{a,t})$  has a perplexity of 41.06.

## 2.5 Brief description of the code

The user simulation prototype for the TownInfo domain has been implemented as a Java class (`NLGUserSimulation.java`). It loads text files with user probabilities generated by CMU SLU toolkit and stores it in a table-look-up (implemented as `java.HashMap`). We use the best performing discounting models for each n-gram to generate the input files.

There are two public methods to access the probabilities. An example of typical use can be found in Listing 1.

- The method `probabilityUserAct(action, history)` takes two strings as parameters. The first one specifies the user action (e.g. `user, select`) and the second the history (e.g. `system IP strategy = system, summaryCompare`). It returns the probability  $P(action|history)$ .
- The method `distributionUserAct((String) history)` takes a specific history as an argument (e.g. `system IP strategy system, summaryCompare`) and returns the distribution over all possible user acts given that history (e.g. `select=0.319149, requestMoreInfo=0.425532, hangup=1.32519E-4, silence=1.47243E-5, askRepeat=2.65037E-4, addInfo=0.234043, other=1.76691E-4`).

Listing 1: Typical use of `NLGUserSimulation`

```
NLGUserSimulation us = new NLGUserSimulation();
us.initusersim(SLUtoolkitFile.txt);
String history = "system,summaryCompare";
string action="user,addInfo"
HashMap<String, Double> distributionUserAct =
    us.callusersimDistribution((String) history);
Double probabilityUserAct =
    us.callusersimAction((String) action, (String) history);
```

The user simulation prototype can be easily integrated in any Reinforcement Learning environment for NLG strategy training. For example, we use the Reinforcement Learning toolkit REALL [Lemon et al., 2006], where we generate a user act by sampling from the distribution of user acts in a specific context (i.e. by calling the user simulation with `us.callusersimDistribution((String) history)`). The prototype can also be used in an interactive mode for demonstration purposes.

## 3 A Two-tier User Simulation Model for the Self-Help domain

### 3.1 Purpose of the user simulation

This section presents a new user simulation model for learning adaptive referring expression generation (REG) policies for spoken dialogue systems in the Self-Help domain. An adaptive REG policy allows a dialogue system to dynamically modify its utterances in order to adapt to user's domain knowledge level. For instance, to refer to the domain objects, the system might use simple descriptive expressions with novices and technical jargon with experts. Such adaptations help grounding between the dialogue partners [Issacs and Clark, 1987]. Since the user's knowledge level is unknown, the system must be able to adapt dynamically during the conversation. Hand-coding such a policy could be extremely difficult. In [Janarthanam and Lemon, 2009c] we have shown that such policies can be learned using simulation based Reinforcement Learning (RL) methods. The quality of such learned policies is directly dependent on the performance of the user simulations used to train them. In our previous work in the Self-Help domain, we employed hand-coded user simulations. We now present a data driven two-tier user simulation model trained on dialogue data collected from real users. We also show that the two-tier model simulates real users more reliable than other data driven baseline n-gram models [Eckert et al., 1997].

### 3.2 Data

We collected dialogue data from 17 participants using a WoZ setup in the Self-Help domain where the task is to help the user to set up his modem, as described in Deliverable 6.1.1. [Boidin et al., 2009] and in [Janarthanam and Lemon, 2009b]. In total the corpus contains ca. 450 system turns. These data contain user reactions to system REG decisions. The strategies used were “jargon”, “descriptive” and “tutorial”. In the jargon strategy the system instructs the user using technical terms (e.g. “Plug the broadband filter into the phone socket.”). In the descriptive strategy, it uses descriptive terms (e.g. “Plug the small white box into the square white box on the wall.”). In the tutorial strategy, the system uses both jargon and descriptive terms together. The system provides clarifications on referring expressions when users request them. Please refer to [Janarthanam and Lemon, 2009b, Boidin et al., 2009] for a more details on our Wizard-of-Oz environment for data collection.

### 3.3 Method

The dialogue data and knowledge profiles collected from the participants were used to build user simulation models. These models take as input the system's dialogue act  $A_{s,t}$  and choice of referring expressions  $REC_{s,t}$  and output the user's dialogue act  $A_{u,t}$  and environment act  $EA_{u,t}$ , i.e. the user interacting with the physical modem.

### 3.3.1 Advanced n-gram model

A simple approach to model real user behaviour is to model user responses (dialogue act and environment act) on many context variables - all referring expressions used in the utterance, the user's knowledge of the REs, history of clarification requests on the REs, and the system's dialogue act. With such a large context, the advanced n-gram model [Georgila et al., 2006a] is ideally the probability distribution found in the real user data.

$$P(A_{u,t} | A_{s,t}, REC_{s,t}, DK_u, H)$$

$$P(EA_{u,t} | A_{s,t}, REC_{s,t}, DK_u, H)$$

However, with such complex contexts, there are data sparsity problems because very many contexts are not seen in the small amount of collected WoZ data.

### 3.3.2 A Two-tier model

Instead of such a complex context model, we could *backoff* and use smaller subcontexts derived from it. We propose a two-tier model, in which the simulation of a user's response is divided into two steps. In the first step, the simulation processes all the referring expressions used by the system  $REC_{s,t}$ . Unlike the advanced n-gram model, this is done one by one, for each expression  $RE_{s,t}$  separately and not for the whole set  $REC_{s,t}$  at once. It returns a clarification request based on each referring expression  $RE_{s,t}$  used, the user's knowledge of the expression  $DK_{RE,u}$ , and previous clarification requests on the expression  $H_{RE}$  and the system dialogue act  $A_{s,t}$ . The clarification request is highly likely in case of the jargon strategy and less likely in other strategies. Also, if a clarification has already been issued, the user is less likely to issue another request for clarification. In such cases, the clarification request model returns *none*.

$$P(CR_{u,t} | RE_{s,t}, DK_{RE,u}, H_{RE}, A_{s,t})$$

In the next step, the model returns a user dialogue act  $A_{u,t}$  and an environment act  $EA_{u,t}$  based on the system dialogue act  $A_{s,t}$  and the clarification request  $CR_{u,t}$ .

$$P(A_{u,t} | A_{s,t}, CR_{u,t})$$

$$P(EA_{u,t} | A_{s,t}, CR_{u,t})$$

By dividing the rich context into smaller subcontexts between the two steps, the two-tier model simulates real users in contexts that are not directly observed in the dialogue data. The model will therefore respond to system utterances containing a mix of REG strategies (e.g. one jargon and one descriptive expression in the same utterance).

### 3.3.3 Baseline Bigram model

A bigram model was built using the dialogue data by conditioning the user responses only on the system's dialogue act [Eckert et al., 1997].

$$P(A_{u,t}|A_{s,t})$$

$$P(EA_{u,t}|A_{s,t})$$

Since it ignores all the context variables except the system dialogue act, it can be used in contexts that are not observed in the dialogue data.

### 3.3.4 Trigram model

The trigram model is similar to the bigram model, but with the previous system dialogue act  $A_{s,t-1}$  as an additional context variable.

$$P(A_{u,t}|A_{s,t},A_{s,t-1})$$

$$P(EA_{u,t}|A_{s,t},A_{s,t-1})$$

### 3.3.5 Equal Probability model baseline

The equal probability model is similar to the bigram model, except that it is not trained on the dialogue data. Instead, it assigns equal probability to all possible responses for the given system dialogue act.

### 3.3.6 Smoothing

We used Witten-Bell discounting [Witten and Bell, 1991] to smooth all our models except the equal probability model, in order to account for unobserved but possible events in dialogue contexts. Witten-Bell discounting extracts a small percentage of probability mass, i.e. number of distinct events observed for the first time ( $T$ ) in a context, out of the total number of instances ( $N$ ), and redistributes this mass to unobserved events in the given context ( $V - T$ ) (where  $V$  is the number of all possible events). The discounted probabilities  $P^*$  of observed events ( $C(e_i) > 0$ ) and unobserved events ( $C(e_i) = 0$ ) are given below.

$$P^*(e_i) = \frac{C(e_i)}{N+T} \text{ if } (C(e_i) > 0)$$

$$P^*(e_i) = \frac{t}{(N+T)(V-T)} \text{ if } (C(e_i) = 0)$$

Model	$A_{u,t}$	$EA_{u,t}$
Two-tier	0.078	0.018
Bigram	0.150	0.139
Trigram	0.145	0.158
Equal Probability	0.445	0.047

Table 2: KL Dialogue Similarity with Modified Witten-Bell discounting

On analysis, we found that the Witten-Bell discounting assigns greater probability to unobserved events than to observed events, in cases where the number of events per context is very low. For instance, in a particular context, the possible events, their frequencies and their original probabilities were - `provide_info` (3, 0.75), `other` (1, 0.25), `request_clarification` (0, 0). After discounting, the revised probabilities  $P^*$  are 0.5, 0.167 and 0.33 respectively. `request_clarification` gets the whole share of extracted probability as it is the only unobserved event in the context and is more than the `other` events actually observed in the data. This is counter-intuitive for our application. Therefore, we use a modified version of Witten-Bell discounting (given below) to smooth our models, where the extracted probability is equally divided amongst all possible events. Using the modified version, the revised probabilities for the illustrated example are 0.61, 0.28 and 0.11 respectively.

$$P^*(e_i) = \frac{C(e_i)}{N+T} + \frac{T}{(N+T)V}$$

### 3.4 Evaluation

We use the Kullback-Leibler (KL) divergence as a similarity measure, as described in Section 2.4. As discussed in Deliverable D3.5, the Kullback-Leibler metric measures how near a probability is to the dataset and does not capture generalisation well. For the purpose of this deliverable, however, it does serve as a metric to compare the alternative user models discussed above.

We consider the advanced n-gram model to be a realistic model of the corpus, as it takes into account all context variables and is reasonably smoothed to account for unobserved responses. Therefore, we compare the probability distributions of all the other models to the advanced n-gram model using the KL dialogue similarity measure. The results of the evaluation are given in Table 2.

The results show that the two-tier model is much closer (0.078, 0.018) to the real user data than the other models. This is due to the fact that the bigram and trigram models do not take into account factors such as the user's knowledge, the strategy used, and the dialogue history.

One key feature of a good user simulation is that it can generalise and model what users might do and say in situations that do not appear in the training data. We posit that the Two-tiered user simulation goes some way to achieving this by dividing the rich context (all variants of which

are unlikely to appear in the database), into smaller sub-contexts that are more likely to appear in the training data.

### 3.5 Brief description of the code

The implementation is similar to the code described in Section 2.5. We use the extracted probabilities to populate a table-look-up function in Java, which can be accessed by RL tools, such as REALL [Lemon et al., 2006].

## 4 Comparison

We now briefly compare the user simulations built for the different domains. In both domains, we built data-driven, sequential n-gram models using small amounts of WoZ data (450 system turns in the SelfHelp domain and 468 IP turns in TownInfo). The Self-Help user simulation is sensitive to REG choices and varies its replies over 3 user actions (provide info, other, request clarification). The TownInfo user simulation is sensitive to Information Presentation choices (IP strategy, number of attributes, previous user act) and varies its replies over 7 user acts (select, requestMoreInfo, addInfo, askRepeat, silence, hangup, other). Both models use discounting techniques and back-off to overcome data sparsity.

The range of values for the Kullback-Leibler divergence is similar for the two domains: The KL results for the TownInfo user simulations range between  $KL_{TownInfo} = 0.011$  and  $KL_{TownInfo} = 0.5$ , see Section 2.4. The KL results for the Self-Help user simulation range between  $KL_{SelfHelp} = 0.018$  and  $KL_{SelfHelp} = 0.445$ , see Section 3.4.

We conclude that using back-off and smoothing techniques allows us to build user simulations which produce dialogue actions which are close to the ones observed in the original data, given the low KL scores in both domains ( $KL_{SelfHelp} = 0.011$  and  $KL_{TownInfo} = 0.018$ ).

## 5 Conclusion

User simulation is an important research topic in the field of spoken dialogue systems because collecting and annotating real interactions with users is often expensive and time consuming. However, such data is required for designing and assessing dialogue systems – in particular, they are needed for training and evaluating the CLASSiC statistical dialogue managers (see D1.4.2) and statistical Natural Language Generation (see D4.4.1). The general goal of the user simulation is to produce user responses that are natural, varied, and consistent using as little initial data as possible. Another goal of user simulation is to expand existing datasets. This latter goal is difficult to evaluate because some situations have never been seen, so it is difficult to predict how a typical user might behave therein.

Here, we have presented two different data-driven user simulation models for two different applications, illustrating that the techniques can be used easily in different domains. Specifically,

we have presented n-gram user simulations to optimise Information Presentation strategies in the TownInfo domain; and Two-tier user simulations to optimise Referring Expression Generation in the Self-Help domain. We described the methods used to build the user simulations from small amounts of Wizard-Of-Oz data. In both domains the models use back-off and discounting techniques to overcome data sparsity. We evaluated the user simulations using the Kullback-Leibler (KL) divergence to measure dialogue similarity and we observed low KL scores in both domains ( $KL_{SelfHelp} = 0.011$  and  $KL_{TownInfo} = 0.018$ ). In terms of impact of this work, the Two-tier approach is unique to the CLASSiC project and offers a mechanism to capture essential but rarely seen contexts, thus improving generalisability.

## References

- [Ai and Litman, 2006] Ai, H. and Litman, D. (2006). Comparing real-real, simulated-simulated, and simulated-real spoken dialogue corpora. In *Proc. of the AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*.
- [Ai and Litman, 2007] Ai, H. and Litman, D. (2007). Knowledge consistent user simulations for dialog systems. In *Proc. of the International Conference of Spoken Language Processing (Interspeech/ICSLP)*.
- [Ai et al., 2007] Ai, H., Tetreault, J., and Litman, D. (2007). Comparing user simulation models for dialog strategy learning. In *Proc. of the North American Meeting of the Association of Computational Linguistics (NAACL)*.
- [Boidin et al., 2009] Boidin, C., Rieser, V., Janarthanam, S., and Lemon, O. (2009). *Domain-limited TTS corpus for expressive speech synthesis and Wizard-of-Oz Data for NLG Strategies*. Deliverable 6.1.1, CLASSiC Project.
- [Clarkson and Rosenfeld, 1997] Clarkson, P. and Rosenfeld, R. (1997). Statistical Language Modeling Using the CMU-Cambridge Toolkit. In *Proc. of ESCA Eurospeech*.
- [Cuayáhuitl et al., 2005] Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2005). Human-computer dialogue simulation using hidden markov models. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.
- [Eckert et al., 1997] Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. In *Proc. of the IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*.
- [Eckert et al., 1998] Eckert, W., Levin, E., and Pieraccini, R. (1998). Automatic evaluation of spoken dialogue systems. In *Proc. of Formal semantics and pragmatics of dialogue (TWLT13)*.
- [Georgila et al., 2006a] Georgila, K., Henderson, J., and Lemon, O. (2006a). User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of the International Conference of Spoken Language Processing (Interspeech/ICSLP)*.
- [Georgila et al., 2006b] Georgila, K., Henderson, J., and Lemon, O. (2006b). User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of the International Conference of Spoken Language Processing (Interspeech/ICSLP)*.
- [Issacs and Clark, 1987] Issacs, E. A. and Clark, H. H. (1987). References in conversations between experts and novices. *Journal of Experimental Psychology: General*, 116:26–37.
- [Janarthanam and Lemon, 2009a] Janarthanam, S. and Lemon, O. (2009a). A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies. In *Proc. of SIGdial*.

- [Janarthanam and Lemon, 2009b] Janarthanam, S. and Lemon, O. (2009b). A Wizard-of-Oz Environment to study Referring Expression Generation in a Situated Spoken Dialogue Task. In *Proceedings of ENLG*.
- [Janarthanam and Lemon, 2009c] Janarthanam, S. and Lemon, O. (2009c). Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In *Proceedings of ENLG*.
- [Janarthanam and Lemon, 2010] Janarthanam, S. and Lemon, O. (2010). Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In Krahmer, E. and Theune, M., editors, *Empirical Methods in Natural Language Generation*, volume 5980 of *Lecture Notes in Computer Science*. Springer, Berlin / Heidelberg.
- [Jung et al., 2009] Jung, S., Lee, C., Kim, K., Jeong, M., and Lee, G. G. (2009). Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer, Speech & Language*, 23:479–509.
- [Katz, 1987] Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35:400–401.
- [Lemon et al., 2006] Lemon, O., Liu, X., Shapiro, D., and Tollander, C. (2006). Hierarchical Reinforcement Learning of dialogue policies in a development environment for dialogue systems: REALL-DUDE. In *Proc. of the 10th SEMdial Workshop on on the Semantics and Pragmatics of Dialogues*.
- [Liu et al., 2009] Liu, X., Rieser, V., and Lemon, O. (2009). A wizard-of-oz interface to study information presentation strategies for spoken dialogue systems. In *Proc. of the 1st International Workshop on Spoken Dialogue Systems*.
- [Manning and Schuetze, 1999] Manning, C. and Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
- [Ney et al., 1994] Ney, H., Essen, U., and Kneser, R. (1994). On structuring probabilistic dependencies in stochastic language modeling”. *Computer Speech and Language*, 8(1).
- [Rieser and Lemon, 2006] Rieser, V. and Lemon, O. (2006). Cluster-based user simulations for learning dialogue strategies. In *Proc. of the 9th International Conference of Spoken Language Processing (Interspeech/ICSLP)*.
- [Rieser and Lemon, 2009] Rieser, V. and Lemon, O. (2009). Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems. In *Proc. of EACL*.
- [Rieser and Lemon, 2010] Rieser, V. and Lemon, O. (2010). Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems. In Krahmer, E. and Theune, M., editors, *Empirical Methods in Natural Language Generation*, volume 5980 of *Lecture Notes in Computer Science*. Springer, Berlin / Heidelberg.

- [Schatzmann et al., 2005] Schatzmann, J., Georgila, K., and Young, S. (2005). Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. of the 6th SIGdial Workshop on Discourse and Dialogue*.
- [Schatzmann et al., 2007] Schatzmann, J., Thomson, B., and Young, S. (2007). Statistical user simulation with a hidden agenda. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*.
- [Schatzmann et al., 2006] Schatzmann, J., Weilhammer, K., Stuttle, M., and Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*, 21(2):97–126.
- [Scheffler and Young, 2001] Scheffler, K. and Young, S. (2001). Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proc. NAACL Workshop on Adaptation in Dialogue Systems*.
- [Williams, 2007] Williams, J. (2007). A method for evaluating and comparing user simulations: The cramer-von mises divergence. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.
- [Witten and Bell, 1991] Witten, I. H. and Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(4).