Regularized Neural User Model for Goal Oriented Spoken Dialogue Systems

Manex Serras, María Inés Torres and Arantza del Pozo

Abstract User simulation is widely used to generate artificial dialogues in order to train statistical spoken dialogue systems and perform evaluations. This paper presents a neural network approach for user modeling that exploits an encoder-decoder bidirectional architecture with a regularization layer for each dialogue act. In order to minimize the impact of data sparsity, the dialogue act space is compressed according to the user goal. Experiments on the Dialogue State Tracking Challenge 2 (DSTC2) dataset provide significant results at dialogue act and slot level predictions, outperforming previous neural user modeling approaches in terms of F1 score.

1 Introduction

Developing statistical Spoken Dialogue Systems (SDS) requires a high amount of dialogue samples from which Dialogue Managers (DM) learn optimal strategies. As manual dialogue compilation is highly resource demanding, an usual approach is to develop an artificial user or User Model (UM) from a small dataset capable of generating synthetic dialogue samples for training and evaluation purposes [19]. UMs are designed in a way that they receive an input from the DM and return a coherent response. A consistent model is expected to maintain coherence throughout the dialogue and to imitate the behavior of real users. Also, some degree of variability is desired in order to generate unseen interactions.

There have been several user modeling proposals in the literature. Initial ap-

Manex Serras · Arantza del Pozo
HSLT department, Vicomtech-IK4 Research centre, Donostia-San Sebastian, Spain, e-mail: mserras@vicomtech.org, e-mail: adelpozo@vicomtech.org

María Inés Torres
Speech Interactive Research Group, Universidad del País Vasco UPV/EHU, Spain e-mail: manes.torres@ehu.eus

proaches [6, 13, 14] used N-grams to model user behavior, but were not capable of capturing dialogue history and, thus, lacked coherence. Subsequent efforts to induce more coherent UMs were proposed by [20, 17]. However, these methods often required a large amount of hand-crafting to infer the dialogue interaction rules.

Several statistical UM approaches have tried to reduce the amount of manual effort required while maintaining dialogue coherence. In [15] Bayesian Networks were used to explicitly incorporate the user goal into the UM. A network of Hidden Markov Models (HMM) was proposed in [5], each HMM representing a goal in the conversation. A hidden-agenda where the user goal is predefined as an agenda of constraints and pieces of information to request to the system and updated at each dialogue turn was presented in [18]. Other approaches have proposed the use of inverse reinforcement learning [2], exploiting the analogies between user simulation and imitation learning.

Recently, a sequence-to-sequence neural network architecture has been proposed [12] for user modeling. Taking into account the whole dialogue history and the goal of the user, this method predicts the next user action as a sequence decoding of dialogue acts. Despite proven to be a promising approach, it suffers from data sparsity when it comes to represent dialogue acts at slot value level.

This paper proposes to model the user as an ensemble of bidirectional encoder-decoder neural networks. The dialogue history is encoded as a sequence instead of a single vector to avoid the information loss caused by compression [3]. Before the decoding process, an additional layer is used to learn regularization parameters that are applied to the encoded sequence in order to improve the generalization of the model. Each user dialogue act is trained in an independent network and an ensemble is constructed by joining all expert networks to predict the user action for each turn. In order to address the data sparsity problem, both system and user dialogue act representations are compressed at slot value level according to the user goal. This representation allows the slot level information to be included in the network during the training process, and thus, represents the dialogue interaction logic with finer granularity.

The paper is structured as follows: Section 2 introduces goal oriented SDS and explains how dialogue act representations are compressed according to the goals set in the dialogue scenario. Section 3 describes the proposed neural network architecture in detail. Section 4 presents the experiments carried out on the DSTC2 dataset. Finally, Section 5 summarizes the main conclusions and sets guidelines for future work.

2 Compressing Goal Oriented Dialogue Acts

Statistical approaches to dialog management require a large amount of dialog samples to train the involved models. Human-to-human dialogues are generally used to train open domain dialogue systems. However, for goal oriented human-machine interaction a common practice to obtain controlled dialogue samples is to assign the users a scenario to fulfill through the dialogue [10, 8]. Such scenario may contain diverse goals to complete by the user and other relevant information for the upcoming interaction.

As explained in [18], the dialogue goal $G = (C, R)$ can be represented as a set of constraints C to inform and values to request R that the user needs to fulfill through the interaction. Table 1 shows a dialogue scenario given in the Dialogue State Tracking Challenge 2 (DSTC2) corpus used in the experimental section of this paper, where $C = (food = international, area = north)$ and $R = (address, phone)$ are given explicitly. The constraints and requests of the goal have a direct correlation with the user's intention at semantic level. User-system interactions are usually represented through dialogue acts (DA) [7, 1, 4], denoted by intention tags (e.g. *inform, request, confirm*) which can contain information objects known as slots, with their corresponding values.

Table 1 Example scenario of a restaurant domain corpus and its goal representation

Description:	You are looking for a restaurant in the north part of town and it should serve international food. Make sure you get the address and phone number.
Constraints:	*Food: International* *Area: North*
Requests:	*Address* *Phone*

The following example shows an utterance annotated using the dialogue act schema: *inform* and *request* are the dialogue acts; *food*, *area*, *address* and *phone* are slots while *international* and *north* are slot values. Note that there can be dialogue acts without slots and that not all slots need to have a specific value.

Utterance:	I want a restaurant that serves international food in the north. Give me it's address and phone number.
DA Representation:	*inform (food=international, area=north) & request (address, phone)*

The main problem of the dialogue act representation is the huge amount of possible slot values. For example, in the DSTC2 corpus, the system can inform of more than 90 food values and 100 restaurant names. Assuming the slot values returned by the system are relevant to the user only if they match a constraint set in the dialogue scenario, every slot value can be replaced with an *is_goal* or *not_goal* token, depending on whether or not they match the given constraint values. Following this assumption, the possible values of the food slot can be reduced from more than 90 to only two.

Table 2 shows how the slot values of an interaction represented at dialogue act level are compressed according to the user goal constraints given in Table 1. As it can be seen, this assumption has a direct impact in the dialogue act representation schema, narrowing down each slot to just two values. As a result, slot value level information can be included in dialogue act representations for user modeling purposes, avoiding excessive sparsity and with small information loss.

Table 2 Dialogue act interaction example, compressed according to the user goal

Original Interaction	**Compressed Interaction**
System: request(area)	System: request(area)
User: inform(area = north , food = international)	User: inform(area = is_goal , food = is_goal)
System: expl-conf(food = italian)	System: expl-conf(food = not_goal)
User: negate( )&inform(food = international)	User: negate( )&inform(food = is_goal)
System: offer(restaurant)&inform(area = north, food=international)	System: offer(restaurant)&inform(area = is_goal, food=is_goal)
User: request(address, phone)	User: request(address, phone)

3 Regularized Bi-directional LSTM User Model

The neural network architecture proposed for user modeling is a bidirectional encoder-decoder with a regularization layer. It encodes the dialogue history in a sequence both forward and backward and exploits a regularization mechanism to set the focus only on the relevant sections of the encoded sequence.

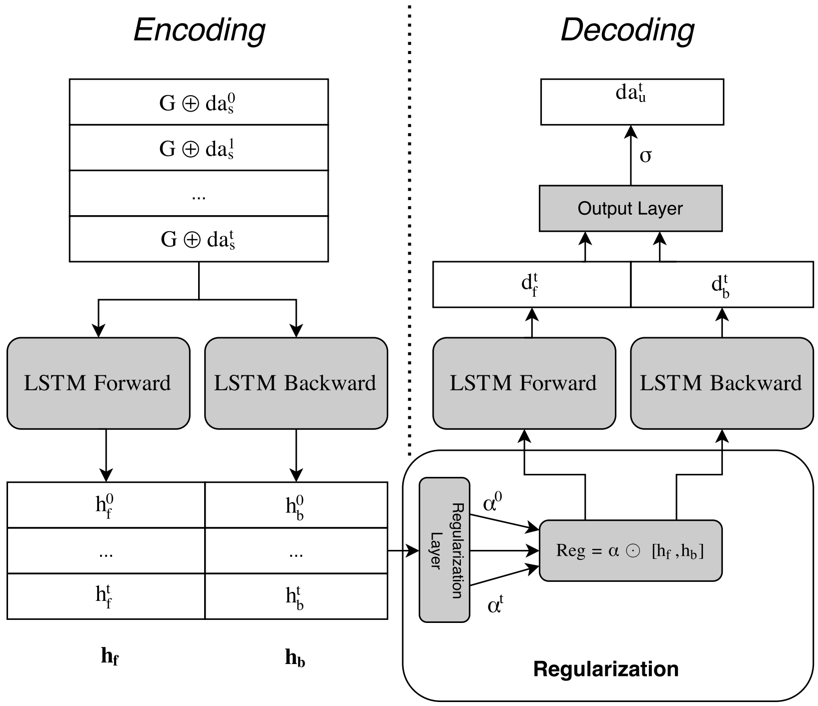

Fig. 1 Neural network architecture proposed for user modeling

As shown in Fig 1, the input to the network is a concatenation of the user goal set in the dialogue scenario G and the sequence of system dialogue acts until the current turn t, $DA_s^t = (da_s^0, da_s^1, ..., da_s^t)$. G is represented as a 1-hot encoding of the slots given as constraints and requests in the dialogue scenario. The output of the network is a prediction of the user dialogue act at the current turn da_u^t. Note that while system dialogue acts change turn by turn, the initial goal representation remains the same throughout the dialogue.

The encoding layer is composed of a bidirectional Long Short Term Memory (LSTM) [9], whose output is the dialogue history encoded as $\mathbf{h_f}$ forward and as $\mathbf{h_b}$ backward.

The applied regularization mechanism requires to learn the weight vectors α for each row of the encoding matrix $H = [\mathbf{h_f}, \mathbf{h_b}]$. Being H_i the i-th row of the encoding matrix, the vector α_i is calculated as $\alpha_i = \sigma(W_a H_i)$, where W_a are the parameters of the Regularization Layer and σ the sigmoid function. Once H and α are known, the encoded sequence is regularized by the element-wise product as follows: $Reg = \alpha \odot H$. This operation will override the non-relevant values of the encoded sequence .

Decoding is then applied to Reg through another bidirectional LSTM, which outputs forward and backward decoding vectors d_f^t and d_b^t at turn t. These vectors are finally concatenated and processed by the output layer with a sigmoid activation function, from which the user dialogue act at the current turn da_u^t is predicted.

The proposed model uses an expert network for every possible user dialogue act, so the architecture in Fig 1 is replicated for each dialogue act of the user. As a result, the final user model is an ensemble of networks, each of which predicts the slots of a specific user dialogue act as shown in Fig 2. A dialogue act is triggered when the corresponding network of the ensemble returns any value above an individual threshold θ set in the development phase (e.g. θ_{inform} for the *inform* dialogue act). The final output is the combination of dialogue acts given by the ensemble of neural networks.

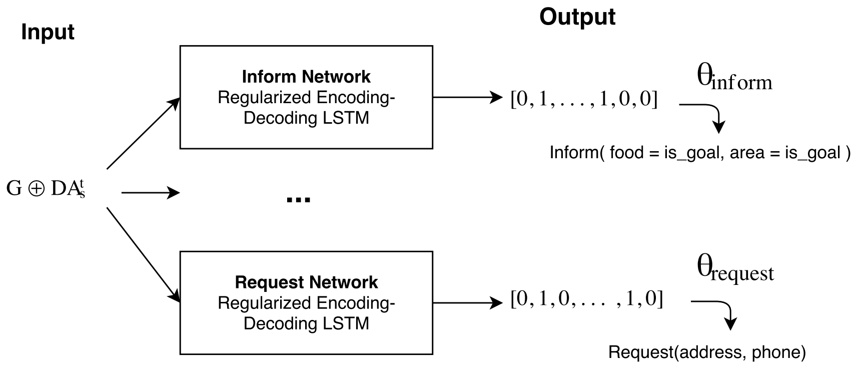

Fig. 2 Ensemble of Dialogue Act networks from which da_u^t is predicted

4 Experimental Framework

4.1 Dialogue State Tracking Challenge 2

The presented neural user model has been tested on the Dialogue State Tracking Challenge 2 (DSTC2) corpus. The second edition of the DSTC series [8] was focused on tracking the dialogue state of a SDS in the Cambridge restaurant domain. For such purpose, a corpus with a total of 3235 dialogues was released [1]. Amazon Mechanical Turk was used to recruit users who would interact with a spoken dialogue system. Each user was given a scenario similar to that described in Table 1, which had to be completed interacting with the system. The goals defined in such scenarios followed the agenda approach of [18]. As a result, the constraints and requests of the user goal are explicitly annotated for each dialogue in the corpus. Table 3 summarizes the user dialogue acts of the DSTC2 corpus with their slots. Note that many dialogue acts do not have related slots and that the slots of the *Request* dialogue act have no value. Table 4 include all the informable slots in the DSTC2 corpus and some examples of their possible values.

Table 3 Dialogue acts that the user can trigger in the DSTC2 corpus

User Dialogue Act	Related Slots
Acknolwedge	*Null*
Affirm	*Null*
Bye	*Null*
Confirm	*Area, Food, Price Range, Restaurant*
Deny	*Area, Food, Price Range, Restaurant*
Hello	*Null*
Help	*Null*
Inform	*Area, Food, Price Range, Restaurant*
Negate	*Null*
Repeat	*Null*
Request Alternatives	*Null*
Request More	*Null*
Request	*Area, Food, Price Range, Restaurant,* *Phone, Address, Signature, Postcode*
Restart	*Null*
Silence	*Null*
Thankyou	*Null*

[1] http://camdial.org/∼ mh521/dstc/

Table 4 Possible slot values for the *Inform, Confirm* and *Deny* dialogue acts

Informable Slots	**Possible Values**	**Examples**
Restaurant Name	113	Nandos, Pizza Hut, ...
Food Type	91	Basque, Italian, European, ...
Price Range	3	Cheap, Moderate, Expensive
Area	5	North, west, south, east, centre

The train/development/test set partitions of the corpus have been used to train, validate and test the proposed methodology. The development set has been used to set the thresholds ($\theta_{inform}, \cdots, \theta request$) and to control overfitting based on early stopping. The test set has been used to carry out final evaluation in terms of Precision, Recall and F1-score as in [12, 19, 5, 16]. These metrics allow comparing the dialogue acts of real and simulated users, measuring the behavior and consistency of the model.

4.2 Experiments and Results

This section describes how the ensemble of networks was trained on the DSTC2 corpus and shows the results achieved, both at dialogue act and slot levels.

Training was done using mini-batch learning; having a dialogue N turns, the total batch is of size N and each input is the sequence of system dialogue acts until turn $t \leq N$. For gradient descent, the Adam [11] optimization method was used with a fixed step size of 0.001. No dropout nor weight penalties were employed. The loss function was computed using the squared error for multiple slot output dialogue acts (e.g. *inform, request*) and cross-entropy for single output dialogue acts (*bye, acknowledge).* Each layer of every dialogue act network had 256 neurons. The individual threshold θ for each dialogue act network ($\theta_{inform}, \cdots, \theta_{bye}$) is set using the development set. In order to set the threshold's value for each dialogue act, a grid search is done to maximize the individual F1 score.

Table 5 shows the Precision, Recall and F1 score achieved by the proposed model with and without regularization on the DSTC2 development and test sets at dialogue act level. For comparative purposes, results presented by [12] in the first reported neural user modeling approach are included, which outperformed previous bigram and agenda-based approaches. As it can be seen, the proposed method significantly improves the F1 score of the simulated user model. The improvement is justified by the increase of overall network complexity and the exploitation of compressed slot value level information for user dialogue act prediction. Also, the regularization mechanism slightly improves the generalization capability of the user model, so its regularized version has been used in the rest of the experiments.

Table 5 Overall results at dialogue act level

		Bi-directional LSTM	**Regularized Bi-directional LSTM**	**Sequence-to-one** [12]
DSTC2 Dev	*Precision*	0.69	0.70	-
	Recall	0.71	0.72	-
	F1	0.70	**0.71**	0.37
DSTC2 Test	*Precision*	0.68	0.71	-
	Recall	0.71	0.73	-
	F1	0.69	**0.72**	0.29

Table 6 summarizes the results achieved for each user dialogue act. As it can be seen from the table, the proposed simulated user is capable of modeling high frequency dialogue acts with ease, but struggles when it comes to low frequency ones. Despite a neural network is trained using the whole corpus for each dialogue act, there are some cases where there is still not enough data to make any prediction.

Table 6 Results for each user dialogue act

	Dev. set					**Test Set**				
Speech Act	**Apparison**	**Predicted**	**Prec**	**Rec**	**F1**	**Apparison**	**Predicted**	**Prec**	**Rec**	**F1**
Ack	0	0	0	0	0	9	0	0	0	0
Affirm	144	129	0.73	0.65	0.69	601	677	0.70	0.79	0.75
Bye	526	528	0.82	0.82	0.82	1169	1082	0.85	0.78	0.81
Confirm	39	0	0	0	0	46	0	0	0	0
Deny	4	0	0	0	0	4	0	0	0	0
Hello	18	0	0	0	0	39	0	0	0	0
Inform	1647	1696	0.80	0.82	0.81	4685	4456	0.85	0.82	0.83
Negate	68	62	0.51	0.47	0.49	261	217	0.46	0.38	0.42
Null	385	480	0.15	0.19	0.17	746	1335	0.10	0.18	0.13
Repeat	7	0	0	0	0	8	0	0	0	0
Reqalts	275	326	0.37	0.44	0.40	649	842	0.36	0.47	0.41
Reqmore	1	0	0	0	0	1	0	0	0	0
Request	1043	1093	0.77	0.81	0.78	2243	2299	0.80	0.82	0.81
Restart	3	0	0	0	0	7	0	0	0	0
Thankyou	510	531	0.79	0.82	0.81	1125	1101	0.80	0.78	0.79

Table 7 shows overall results achieved at slot value level. As expected, performance decreases given the finer granularity of the task but still remains high considering the extra complexity involved.

Table 7 Overall results at slot value level

Dataset	**Precision**	**Recall**	**F1**
DSTC2 Dev	0.60	0.63	0.62
DSTC2 Test	0.60	0.64	0.62

Finally, Tables 8 and 9 show a more exhaustive evaluation of the two dialogue acts with highest impact on the DSTC2 corpus: *Inform* and *Request*, at compressed slot value and slot level respectively.

Table 8 *Inform* dialogue act results at compressed slot value level

		Development Set					**Test Set**				
Inform acts	**Slot value**	**Count**	**Predicted**	**Prec**	**Rec**	**F1**	**Count**	**Predicted**	**Prec**	**Rec**	**F1**
Food	*is_goal*	509	719	0.54	0.76	0.63	1472	1970	0.60	0.80	0.69
	not_goal	299	204	0.39	0.26	0.31	809	775	0.45	0.43	0.44
Area	*is_goal*	423	454	0.67	0.72	0.70	1071	1042	0.70	0.68	0.69
	not_goal	21	0	0	0	0	115	0	0	0	0
Price range	*is_goal*	375	342	0.75	0.69	0.72	908	833	0.72	0.67	0.70
	not_goal	16	0	0	0	0	116	0	0	0	0
This	*Don't care*	231	308	0.61	0.81	0.70	767	898	0.59	0.69	0.64

As it can be seen in the table, the simulated user achieves good results informing slot values set as goals in the initial scenario, but suffers from heavy degradation when it comes to inform those that are not defined as such.

Table 9 *Request* dialogue act results at slot level

	Development Set					**Test Set**				
Requested Slot	Count	Predicted	Prec	Rec	F1	Count	Predicted	Prec	Rec	F1
Food	92	56	0.66	0.40	0.5	134	140	0.51	0.53	0.52
Area	40	21	0.71	0.38	0.49	113	36	0.75	0.24	0.36
Pricerange	90	62	0.64	0.44	0.52	115	93	0.62	0.50	0.56
Address	421	549	0.62	0.81	0.70	939	1154	0.68	0.84	0.75
Phone	426	498	0.64	0.75	0.69	986	999	0.72	0.73	0.72
Postcode	94	80	0.75	0.64	0.68	219	163	0.83	0.61	0.70
Signature	2	0	0	0	0	10	0	0	0	0

In relation to the behavior of the user model with regard to the *Request* dialogue act, the correlation between high F1 scores and the requested slot occurrence is clear; the higher the slot occurrence, the better the F1 score.

5 Conclusions and future work

This paper has presented a neural user model for goal oriented spoken dialogue systems. The proposed approach employs a sensible way to exploit slot level information without adding unnecessary sparsity to the representation. The ensemble of bidirectional encoding-decoding networks is capable of exploiting the full dialogue history efficiently and the regularization technique slightly improves the generalization capability of the model. These changes provide significant results both at dialogue act and slot level predictions, outperforming previous neural user modeling approaches in terms of F1 score.

Future work will require refining the presented architecture, so that it can model low-occurrence dialogue acts and slot values more precisely. The approach should also be tested on additional goal oriented dialogue datasets. The proposed simulated user model will be compared against other user modeling approaches, when it comes to training and evaluating statistical spoken dialogue systems. In addition, having real users evaluate the generated policies shall provide useful insights about the modeling capabilities of the network.

References

1. Nicholas Asher and Alex Lascarides. Indirect speech acts. *Synthese*, 128(1):183–228, 2001.
2. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. User simulation in dialogue systems using inverse reinforcement learning. In *Interspeech 2011*, pages 1025–1028, 2011.
3. Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.
4. Mark G Core and James Allen. Coding dialogs with the damsl annotation scheme. In *AAAI fall symposium on communicative action in humans and machines*, volume 56. Boston, MA, 1997.
5. Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. Human-computer dialogue simulation using hidden markov models. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 290–295. IEEE, 2005.
6. Wieland Eckert, Esther Levin, and Roberto Pieraccini. User modeling for spoken dialogue system evaluation. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 80–87. IEEE, 1997.
7. Michael Hancher. The classification of cooperative illocutionary acts. *Language in society*, pages 1–14, 1979.
8. Matthew Henderson, Blaise Thomson, and Jason Williams. Dialog state tracking challenge 2 & 3 handbook. *camdial. org/mh521/dstc*, 2013.
9. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
10. Lluís F. Hurtado, David Griol, Emilio Sanchis, and Encarna Segarra. *A Statistical User Simulation Technique for the Improvement of a Spoken Dialog System*, pages 743–752. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
11. Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pages 1–13, 2015.
12. El Asri Layla, He Jing, and Kaheer Suleman. A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Interspeech*, 2016.
13. Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23, 2000.
14. Olivier Pietquin. *A framework for unsupervised learning of dialogue strategies*. Presses univ. de Louvain, 2005.
15. Olivier Pietquin and Thierry Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):589–599, 2006.

16. Silvia Quarteroni, Meritxell González, Giuseppe Riccardi, and Sebastian Varges. Combining user intention and error modeling for statistical dialog simulators. In *INTERSPEECH*, pages 3022–3025, 2010.
17. Verena Rieser and Oliver Lemon. Cluster-based user simulations for learning dialogue strategies. In *Interspeech*, 2006.
18. Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics, 2007.
19. Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(02):97–126, 2006.
20. Konrad Scheffler and Steve Young. Probabilistic simulation of human-machine dialogues. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, volume 2, pages II1217–II1220. IEEE, 2000.