

Training a BN-based user model for dialogue simulation with missing data

Stéphane Rossignol^{†,‡}, Olivier Pietquin^{†,‡}, Michel Ianotto^{†,‡}

[†]SUPELEC - IMS Research Group

2 rue Édouard Belin, Metz

France

[‡]UMI 2958 (GeorgiaTech - CNRS)

{forename.surname}@supelec.fr

Abstract

The design of a Spoken Dialogue System (SDS) is a long, iterative and costly process. Especially, it requires test phases on actual users either for assessment of performance or optimization. The number of test phases should be minimized, yet without degrading the final performance of the system. For these reasons, there has been an increasing interest for dialogue simulation during the last decade. Dialogue simulation requires simulating the behavior of users and therefore requires user modeling. User simulation is often done by statistical systems that have to be tuned or trained on data. Yet data are generally incomplete with regard to the necessary information for simulating the user decision making process. For example, the internal knowledge the user builds along the conversation about the information exchanged while interacting is difficult to annotate.

In this contribution, we propose the use of a previously developed user simulation system based on Bayesian Networks (BN) and the training of this model using algorithms dealing with missing data. Experiments show that this training method increases the simulation performance in terms of similarity with real dialogues.

1 Introduction

The design of a Spoken Dialogue System (SDS) is a long, iterative and costly process. Although several attempts exist to simplify this design such as the VoiceXML language (W3C, 2008), graphical interfaces (McTear, 1998) or machine-learning-based methods (Pietquin and Dutoit, 2003), it remains an expert job. Especially, it requires test

phases on actual users either for assessment of performance (Eckert et al., 1997; López-Cózar et al., 2006) or strategy optimization by means of reinforcement learning (Levin et al., 1997; Pietquin and Dutoit, 2006a). The number of test phases should be minimized, yet without degrading the final performance of the system. One solution to this problem is the use of Wizard-of-Oz methods (Kelley, 1984; Rieser, 2008). Although this doesn't require a real implementation of the dialogue system to be tested, this is still time and money consuming. For these reasons, there has been an increasing interest for dialogue simulation during the last decade (Eckert et al., 1997; Pietquin and Dutoit, 2006a; Schatzmann et al., 2006; López-Cózar et al., 2006). Dialogue simulation requires simulating the behavior of users and therefore requires user modeling as well as error modelling (Pietquin and Dutoit, 2006b; Schatzmann et al., 2007b). Most often, dialogue simulation takes place at the intention level (Eckert et al., 1997; Pietquin and Dutoit, 2006a; Schatzmann et al., 2007c) but can take place at the speech signal level (López-Cózar et al., 2006). This paper focuses on the former solution and more specifically on statistical user simulation (Eckert et al., 1997; Cuayáhuitl et al., 2005; Pietquin and Dutoit, 2006a; Schatzmann et al., 2007c). Statistical models are generally parametric generative models where parameters are conditional probabilities that can either be hand-tuned (estimated by experts) because of the complexity of the model (Pietquin, 2006; Schatzmann et al., 2007a), trained on actual man-machine dialogue data (Eckert et al., 1997; Cuayáhuitl et al., 2005; Pietquin et al., 2009; Syed and Williams, 2008) or a mix of both (Scheffler and Young, 2001; Keizer et al., 2010) so as to deal with parameters which are not directly accessible in a database. Indeed, data are often incomplete with regard to the necessary information for simulating the user decision making process. For

example, the internal knowledge the user builds along the conversation about the dialogue context is difficult to annotate.

In this contribution, we propose the use of a previously developed user simulation system based on Bayesian Networks (BN) described in Section 2 and the training of this model using algorithms dealing with missing data. As said before, in the case of man-machine dialogues data, some information is often missing in the annotations. This paper focuses on the user’s internal representation of the dialogue context which is referred to as the *knowledge* of the user. This is a major difference with other papers of the literature such as (Syed and Williams, 2008) where transition probabilities are estimated according to the history of system and user acts. Taking into account the incremental knowledge of the user about previous exchanges is important to ensure the consistency of the dialogue during the interaction (Pietquin, 2006). Although it is a difficult task, the knowledge of the user could be inferred from the data itself, by a human expert, a set of rules, or a trained classification algorithm dedicated to this task. In Section 4, this approach is followed, the knowledge (or an accurate estimate) is supposed to be known and the derived training methods for learning the BN parameters are explained. Alternatively, the knowledge of the user can be treated as hidden and the BN parameters can be learned using corresponding Expectation-Maximization algorithms. This approach is described in Section 5, both within a statistical framework (expected-likelihood maximization) and within a Bayesian framework (starting from some prior distribution over parameters). The experiments described in Section 6 show that this training method increases the simulation performance in terms of similarity with real dialogues.

2 BN-based user simulation

The user simulation method studied in this paper is based on the probabilistic model of a man-machine dialog proposed in (Pietquin, 2005; Pietquin and Dutoit, 2006a). The interaction between the user and the dialog manager is seen as a sequential transfer of intentions thanks to dialog acts organized in turns noted t . At each turn t the dialog manager selects a system act a_t conditionally to its internal state s_t and according to its strategy. The user answers by a user act u_t which is

conditioned by the goal g_t s/he is pursuing and the knowledge k_t s/he has about the dialog (what has been exchanged before reaching turn t). So, at a given turn, the information exchange can be modeled thanks to the joint probability $p(a, s, u, g, k)$ of all these variables. This joint probability can be factored as:

$$p(a, s, u, g, k) = p(u|g, k, a, s)p(g|k, a, s)p(k|s, a)p(a|s)p(s)$$

Given that :

- since the user doesn’t have access to the SDS state, u, g and k cannot depend on s ,
- the user’s goal can only be modified according to his/her knowledge of the dialog,

this expression can be simplified:

$$p(a, s, u, g, k) = \underbrace{p(u|g, k, a)}_{\text{User act}} \underbrace{p(g|k)}_{\text{Goal Modif.}} \underbrace{p(k|a)}_{\text{Know. update}} \underbrace{p(a|s)}_{\text{DM Policy}} p(s)$$

This can be expressed by the Bayesian network depicted on Fig. 1.

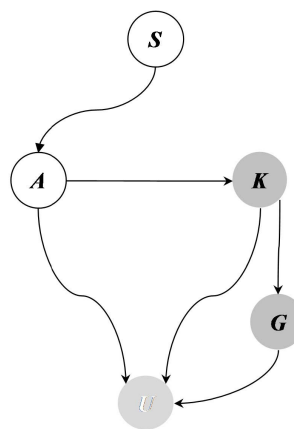


Figure 1: Bayesian Network-based Simulated User

As explained in (Pietquin and Dutoit, 2006a), the practical use of this kind of BN requires a tractable representation of the stochastic variables $\{a, s, u, g, k\}$. Variables are therefore considered as vectors of either boolean either symbolic values which makes them discrete in any case and limits the number of conditional probabilities which are the parameters Θ of this model (see (Pietquin, 2005; Pietquin and Dutoit, 2006a) for more details).

In this BN, nodes represented as empty circles \circ are input variables of the simulator (actually the outputs of the SDS), the light-gray circle \bullet is the output variable (user’s utterance) and the dark-gray circles \bullet are internal variables (knowledge and goal). The parameters linked to these internal variables are generally hard to obtain from data collection since it is hard to infer what are the user knowledge and goal from the transcripts of interactions. The most difficult is probably to infer the knowledge of the user since in data collection settings or Wizard-of-Oz experiments, the user goal can be imposed on test users while the knowledge is built during the interaction and depends on it.

3 Dialogue example within the TownInfo domain

A dialogue example between the simulated user and the HIS dialogue manager (Young et al., 2009) is shown below. The user has the goal to get the name, phone number and address of a restaurant that serves Italian food and beer. The dialogue shows both system (SYS) and user (USR) acts. At the start of the dialogue, the simulator is initialised with a (randomly generated) user goal. Some of the slots are constrained by the user (G.C) and other are requested by him (G.R).

```
G_R> name, phone, addr
G_C> type=restaurant, food=italian, drinks=beer

Sys> hello() [Greet]
Usr> hello() {1.00}
.
Sys> repeat() [Repeat]
Usr> hello(type=restaurant) {1.00}
--
Sys> request(near) [Request]
Usr> inform(food=italian,drinks=beer) {1.00}
--
Sys> inform(name=None,type=restaurant,
            food=italian) [Offer]
Usr> reqalts(food=indian) {1.00}
--
Sys> inform(name="Carnaby Street",type=restaurant,
            drinks=beer,food=indian) [Offer]
Usr> deny(drinks=wine,drinks=beer) {1.00}
--
Sys> inform(name="The Parrot",type=bar,drinks=beer,
            food=indian) [Offer]
Usr> request(phone) {1.00}
--
Sys> inform(name="The Parrot",
            phone="728-66-32") [Inform]
Usr> request(addr) {1.00}
--
Sys> inform(name="The Parrot",
            addr="Art Square") [Inform]
Usr> bye()
--
End> --
```

In the next sections, the training methods used for estimating the parameters of the BN-based user simulation described in Section 2 are described.

4 Training methods with complete data

4.1 Maximum likelihood

When all variables in a dataset are observed, a statistical framework can be used, in which the frequencies of events appearing in the database are computed. This is known as the *maximum likelihood* approach:

$$\Theta_{i,j,k}^{ML} = \hat{p}(X_i = x_k \mid pa(X_i) = x_j) = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}}$$

where the set of $\Theta_{i,j,k}^{ML}$ are the BN parameters that need to be learned, $N_{i,j,k}$ is the number of events in the database for which the variable X_i is in the state x_k and its parents in the network (pa) in the configuration x_j .

4.2 Bayesian training

Bayesian estimation of the parameters is slightly different. It actually aims at estimating the probability distribution over parameters and estimates the parameters using either a *maximum a posteriori* (MAP) approach or the parameters’ expectation given this distribution. This is done knowing that the variables have been observed and requires some prior on the parameters. Using a Dirichlet distribution prior (standard choice for multivariate distributions), it is possible to derive an analytical formula for the expected parameters which is similar to the one obtained in the previous section. Using the MAP approach:

$$\Theta_{i,j,k}^{MAP} = \hat{p}(X_i = x_k \mid pa(X_i) = x_j) = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum_k N_{i,j,k} + \alpha_{i,j,k} - 1}$$

where the $\alpha_{i,j,k}$ are the coefficients of the Dirichlet distribution.

Using the *a priori* expectation approach (AEP) instead of the MAP, one gets:

$$\Theta_{i,j,k}^{AEP} = \hat{p}(X_i = x_k \mid pa(X_i) = x_j) = \frac{N_{i,j,k} + \alpha_{i,j,k}}{\sum_k N_{i,j,k} + \alpha_{i,j,k}}$$

4.3 Priors on parameters

The $\alpha_{i,j,k}$ are priors on parameters' distribution (Dirichlet distribution coefficients), as they are set by an expert. It is thus possible to give to these coefficients more or less importance, given the confidence of the expert. This will result in different trained BN/retrained BN user simulators. Fine-tuning the $\alpha_{i,j,k}$ will allow us to get simulators behaving more or less like the human users which produced the database, as shown in Section 6. Of course, if nothing is known (no expert available), a uniform distribution over parameters (all coefficient being equal) can be taken as a prior and the method can still be used.

5 Training methods with missing data

5.1 Expectation-Maximization algorithm

The *Expectation-Maximization* (EM) algorithm (Dempster et al., 1977) allows estimating the BN parameters even when the data corresponding to some of the parameters is missing.

EM is a recursive algorithm applied until convergence as explained hereafter.

Let us assume that:

- $X_\nu = \{X_\nu^{(l)}\}_{l=1\dots N}$ is the set of the N observable data.
- $\Theta^{(t)} = \{\Theta_{i,j,k}^{(t)}\}$ are the estimations of the parameters of the BN at iteration t .

EM is a recursive algorithm, initialized with arbitrary $\Theta^{(0)}$ values, consisting of two steps:

- **Expectation (E)** step: the missing data $N_{i,j,k}$ are estimated, by computing their expectation conditionally to the data and to the current parameter estimates (i.e., to the current distribution estimate):

$$N_{i,j,k}^* = E[N_{i,j,k}] = \sum_{l=1}^N \hat{p}\left(X_i = x_k \mid pa(X_i) = x_j, X_\nu^{(l)}, \Theta^{(t)}\right)$$

This consists in doing inference using the current parameter values, and in replacing the missing values by the probabilities obtained by inference.

- **Maximization (M)** step: replacing the missing $N_{i,j,k}$ by their expected value computed

in the previous step, it is possible to compute the new parameter values $\Theta^{(t+1)}$, using maximum likelihood:

$$\Theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^*}{\sum_k N_{i,j,k}^*}$$

5.2 Expectation-Maximization algorithm and Bayesian training

The EM algorithm can be used within the Bayesian framework as well. In that case, the maximum likelihood estimation used in the **M** step must be replaced by an *a posteriori* maximum. Using the *a posteriori* expectation, one gets:

$$\Theta_{i,j,k}^{(EM)} = \Theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^* + \alpha_{i,j,k}}{\sum_k N_{i,j,k}^* + \alpha_{i,j,k}}$$

6 Experiment

6.1 Dialogue task and data

To test the different training algorithms, the user simulator parameters have been learnt on a database containing 1234 actual man-machine dialogues in the domain of tourist information. The dialogue system is a large-scale application aiming at retrieving information about user's interests in a city (about restaurants, hotels, *etc.*) so as to provide relevant propositions of venues as described in (Keizer et al., 2010). The venues can be of different types such as bar, restaurants and hotels. Each venue is described by a set of features (type of cuisine, location in the city *etc.*). The hierarchical structure of the task makes it relatively complex as well as the high number of slots (13). The data contains transcripts and semantic annotations in terms of dialogue act. The BN-based user simulator has been tested against the HIS Dialogue Manager developed at Cambridge University (Young et al., 2009).

6.2 Training methods

Six training setups for the BN-based user simulator were tested. 1000 dialogues were generated for each configuration after training. The six setups are described below:

- “ori-T-BN”: the knowledge parameters were estimated on the database and the BN parameters were learned using the results by a Maximum Likelihood method ($\Theta_{i,j,k}^{ML}$) (see Section 4).

- “mod-T-BN”: the knowledge parameters were estimated on the database and the BN parameters were learned with a Bayesian learning method (AEP method) and using priors fixed by an expert, reasonably taken into account ($\Theta_{i,j,k}^{AEP}$) (see Section 4).
- “H-BN”: the BN parameters were hand-coded by an expert (Heuristics).
- “mod-T1-BN”: the knowledge was supposed missing and the BN parameters were learned using the database by Bayesian EM and priors fixed by an expert; first version: expert almost not taken into account ($\Theta^{(EM)}$) (see Section 5).
- “mod-T2-BN”: the knowledge was supposed missing and the BN parameters were learned using the database by Bayesian EM and priors fixed by an expert; second version: expert reasonably taken into account ($\Theta^{(EM)}$).
- “mod-T3-BN”: the knowledge was supposed missing and the BN parameters were learned using the database by Bayesian EM and priors fixed by an expert; third version: expert much taken into account ($\Theta^{(EM)}$).

The last three configurations are the most realistic ones.

6.3 Evaluation methods

Four dissimilarity measures have been computed: the Precision, the Recall, the symmetric Kullback-Leibler dissimilarity DS and the average number of turns per dialog (Pietquin and Hastie, 2011).

Precision:

$$P = 100 \times \frac{\text{Correctly predicted actions}}{\text{All actions in simulated response}}$$

Recall:

$$R = 100 \times \frac{\text{Correctly predicted actions}}{\text{All actions in real response}}$$

$$DS(P||Q) = \frac{D_{KL}(P||Q) + D_{KL}(Q||P)}{2}$$

where

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

and where p_i (resp. q_i) is the frequency of dialogue act a_i in the histogram of distribution P (resp. Q)

	ori-T-BN	mod-T-BN	H-BN
Precision:	47.11	50.62	63.63
Recall:	57.89	60.68	53.20
DS :	0.7292	0.6712	0.8803
Nturns/diag:	18.19	15.15	5.283

Table 1: Dissimilarities using the first three BN configurations

	mod-T1-BN	mod-T2-BN	mod-T3-BN
Precision:	63.71	64.60	67.13
Recall:	61.84	63.83	69.27
DS :	0.6674	0.7864	0.5288
Nturns/diag:	7.690	7.980	8.703

Table 2: Dissimilarities using the last three BN configurations

obtained on the database (resp. on the generated data). The simulated dialogues are compared to the dialogues from the database on this basis. Notice that the Precision and the Recall must be as high as possible, the Kullback-Leibler as low as possible and the average number of turns per dialogue as close to the average number of turns per dialogue in the database (which is 8.185).

6.4 Results

The results are provided in Tables 1 and 2. Table 1 clearly indicates that the first configurations do not provide realistic dialogues. Considering the Recall, the DS and the number of turns, the mod-T-BN gives the best results. The fact that ori-T-BN gives bad results indicates that the database is not large enough, and/or that the inferred knowledge is not very accurate. The H-BN was designed to give as short as possible dialogues: this can be seen in the dissimilarity measures.

Table 2 indicates that the training techniques with missing data are efficient, allowing not to use the error-prone (automatic or manual) knowledge inference. Taking the expert information into account allows to improve the performance to some extent, considering the Precision, the Recall and the number of turns per dialogue dissimilarity measures. The DS dissimilarity measure gives more uncertain results.

7 Conclusions

In this paper, the problem of user simulation in spoken dialogue systems is addressed and particularly the training of statistical user simulation systems on actual data. Most often, actual man-machine dialogue corpora annotations do not contain all the required information for simulating the user's decision-making process. For instance, the knowledge of the dialogue context which is incrementally built by the user during the interaction is very difficult to annotate. To tackle this problem, this contribution proposes the use of expectation-maximization algorithms (in a Maximum Likelihood setting or a Bayesian setting) to learn parameters of a BN-based user model. Experimental results show that this method improves significantly the similarity of automatically generated dialogues.

In the future, this user model will be used to train a reinforcement-learning-based dialogue manager so as to optimize the dialogue strategy. Also, the extension of this user simulation technique to other tasks is envisioned. The simulation of the grounding process which is possible thanks to this kind of model (Rossignol et al., 2010) should also benefit from this training method to generate more realistic dialogues. Finally, we want to compare the performance of this user model to newly proposed models such as in (Chandramohan et al., 2011) according to several metrics (Pietquin and Hastie, 2011).

Acknowledgement

The work presented here has been done during the CLASSiC project (Grant No. 216594, www.classic-project.org) funded by the European Commission's 7th Framework Programme (FP7).

References

Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, and Olivier Pietquin. 2011. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*, Florence (Italy), August.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2005)*, pages 290–295.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User Modeling for Spoken Dialogue System Evaluation. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'97)*, pages 80–87.

Simon Keizer, Milica Gašić, Filip Jurčićek, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Parameter estimation for agenda-based user simulation. In *Proceedings of the SIGdial Conference on Discourse and Dialogue (SIGdial 2010)*, Tokyo, Japan, September.

John Kelley. 1984. An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications. *ACM Transactions on Office Information Systems*, 2(1):26–41.

Ester Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning Dialogue Strategies within the Markov Decision Process Framework. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'97)*, December.

Ramón López-Cózar, Zoraida Callejas, and Michael F. McTear. 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artificial Intelligence Review*, 26(4):291–323.

Michael McTear. 1998. Modelling spoken dialogues with state transition diagrams: experiences with the cslu toolkit. In *Proc 5th International Conference on Spoken Language Processing*, pages 1223–1226.

Olivier Pietquin and Thierry Dutoit. 2003. Aided Design of Finite-State Dialogue Management Systems. In *Proceedings of the 4th IEEE International Conference on Multimedia and Expo (ICME 2003)*, volume III, pages 545–548, Baltimore (USA, MA), July.

Olivier Pietquin and Thierry Dutoit. 2006a. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589–599, March.

Olivier Pietquin and Thierry Dutoit. 2006b. Dynamic Bayesian Networks for NLU Simulation with Application to Dialog Optimal Strategy Learning. In *Proceedings of the 31st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume I, pages 49–52, Toulouse (France), May.

Olivier Pietquin and Helen Hastie. 2011. A survey on metrics for the evaluation of user simulations. *Knowledge Engineering Review*. Accepted for Publication.

- Olivier Pietquin, Stéphane Rossignol, and Michel Ianotto. 2009. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1st International Workshop on Spoken Dialogue Systems Technology (IWSDS 2009)*, Irsee (Germany), December. 4 pages.
- Olivier Pietquin. 2005. A Probabilistic Description of Man-Machine Spoken Communication. In *Proceedings of the 5th IEEE International Conference on Multimedia and Expo (ICME 2005)*, pages 410–413, Amsterdam (The Netherlands), July.
- Olivier Pietquin. 2006. Consistent Goal-Directed User Model for Realistic Man-Machine Task-Oriented Spoken Dialogue Simulation. In *Proceedings of the 7th IEEE International Conference on Multimedia and Expo*, pages 425–428, Toronto (Canada), July.
- Verena Rieser. 2008. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*. Ph.D. thesis, Saarland University, Department of Computational Linguistics, Saarbrücken, July.
- Stéphane Rossignol, Olivier Pietquin, and Michel Ianotto. 2010. Grounding Simulation in Spoken Dialog Systems with Bayesian Networks. In G. Geunbae Lee et al., editor, *Proceedings of the International Workshop on Spoken Dialogue Systems (IWSDS 2010)*, volume 6392 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 110–121, Gotemba (Japan), October. Springer-Verlag, Heidelberg-Berlin.
- Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007a. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) with Human Language Technology Conference (HLT 2007)*, Rochester.
- Jost Schatzmann, Blaise Thomson, and Steve Young. 2007b. Error Simulation for Training Statistical Dialogue Systems. In *Proceedings of the International Workshop on Automatic Speech Recognition and Understanding (ASRU'07)*, Kyoto (Japan).
- Jost Schatzmann, Blaise Thomson, and Steve Young. 2007c. Statistical User Simulation with a Hidden Agenda. In *Proceedings of the SIGDial Workshop on Discourse and Dialogue (SIGdial'07)*, Anvers (Belgium).
- Konrad Scheffler and Steve Young. 2001. Corpus-Based Dialogue Simulation for Automatic Strategy Learning and Evaluation. In *Proc. NAACL Workshop on Adaptation in Dialogue Systems*.
- Umar Syed and Jason D. Williams. 2008. Using Automatically Transcribed Dialogs to Learn User Models in a Spoken Dialog System. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) with Human Language Technology Conference (HLT 2008)*, Columbus, Ohio, USA.
- W3C, 2008. *VoiceXML 3.0 Specifications*, December. <http://www.w3.org/TR/voicexml30/>.
- Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Blaise Thomson, and Kai Yu. 2009. The Hidden Information State Model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, April.