Engineering Statistical Dialog State Trackers: A Case Study on DSTC

Daejoong Kim, Jaedeug Choi, Kee-Eung Kim
Department of Computer Science, KAIST
South Korea
{djkim, jdchoi, kekim}@ai.kaist.ac.kr

Jungsu Lee, Jinho Sohn
LG Electronics
South Korea
{jungsu.lee, jinho.sohn}@lge.com

Abstract

We describe our experience with engineering the dialog state tracker for the first Dialog State Tracking Challenge (DSTC). Dialog trackers are one of the essential components of dialog systems which are used to infer the true user goal from the speech processing results. We explain the main parts of our tracker: the observation model, the belief refinement model, and the belief transformation model. We also report experimental results on a number of approaches to the models, and compare the overall performance of our tracker to other submitted trackers. An extended version of this paper is available as a technical report (Kim et al., 2013).

1 Introduction

In spoken dialog systems (SDSs), one of the main challenges is to identify the user goal from her utterances. The significance of accurately identifying the user goal, referred to as *dialog state tracking*, has emerged from the need for SDSs to be robust to inevitable errors in the spoken language understanding (SLU).

A number of studies have been conducted to track the dialog state through multiple dialog turns using a probabilistic framework, treating SLU results as noisy observations and maintaining probability distribution (*i.e.*, belief) on user goals (Bohus and Rudnicky, 2006; Mehta et al., 2010; Roy et al., 2000; Williams and Young, 2007; Thomson and Young, 2010; Kim et al., 2011).

In this paper, we share our experience and lessons learned from developing the dialog state tracker that participated in the first Dialog State Tracking Challenge (DSTC) (Williams et al., 2013). Our tracker is based on the belief update in the POMDP framework (Kaelbling et al., 1998), particularly the hidden information state (HIS) model (Young et al., 2010) and the partition recombination method (Williams, 2010).

2 Dialog State Tracking

Our tracker mainly follows the belief update in HIS-POMDP (Young et al., 2010). The SDS executes system action a, and the user with goal g responds to the system with utterance u. The SLU processes the utterance and generates the result as an N-best list $\boldsymbol{o} = [\langle \tilde{u}_1, f_1 \rangle, \ldots, \langle \tilde{u}_N, f_N \rangle]$ of the hypothesized user utterance $\tilde{u}_i$ and its associated confidence score f_i. Because the SLU is not perfect, the system maintains a probability distribution over user goals, called a *belief*. In addition, the system groups user goals into equivalence classes and assigns a single probability for each equivalence class since the number of user goals is often too large to perform individual belief updates for all possible user goals. The equivalence classes are called partitions and denoted as ψ. Hence, given the current belief b, system action a, and recognized N-best list $\boldsymbol{o}$, the dialog state tracker updates the belief b' over partitions as follows:

$$b'(\psi') \propto \sum_u \Pr(\boldsymbol{o}|u) \Pr(u|\psi', a) \Pr(\psi'|\psi) b(\psi) \tag{1}$$

where $\Pr(\boldsymbol{o}|u)$ is the observation model, $\Pr(u|\psi, a)$ is the user utterance model, $\Pr(\psi'|\psi)$ is the belief refinement model.

2.1 Observation Model

The observation model $\Pr(\boldsymbol{o}|u)$ is the probability that the SLU produces the N-best list $\boldsymbol{o}$ when the user utterance is u. We experimented with the following three models for the observation model.

Confidence score model: as in HIS-POMDP, this model assumes that the confidence score f_i obtained from the SLU is exactly the probability of generating the hypothesized user utterance $\tilde{u}_i$. Hence, $f_i = \Pr(\tilde{u}_i, f_i|u)$.

Histogram model: this model estimates a function that maps the confidence score to the probability of correctness. We constructed a histogram of confidence scores from the training datasets to obtain the empirical probability $\Pr(cor(f_i))$ of whether the entry associated with confidence score f_i is a correct hypothesis or not.

Generative model: this model is a simplified version of a generative model in (Williams, 2008) that only uses confidence score: $\Pr(\tilde{u}_i, f_i|u) = \Pr(cor(i)) \Pr(f_i|cor(i))$ where $\Pr(cor(i))$ is the probability of the i-th entry being a correct hypothesis and $\Pr(f_i|cor(i))$ is the probability of the i-th entry having confidence score f_i when it is a correct hypothesis.

2.2 User Utterance Model

The user utterance model $\Pr(u|\psi, a)$ indicates how the user responds to the system action a when the user goal is in ψ. We adopted the HIS-POMDP user utterance model, consisting of a bigram model and an item model. The details are described in (Kim et al., 2013).

2.3 Belief Refinement Model

Given the SLU result $\tilde{u}_i$ and the system action a, the partition ψ is split into ψ'_i with probability $\Pr(\psi'_i|\psi)$ and $\psi - \psi'_i$ with probability $\Pr(\psi - \psi'_i|\psi)$. The belief refinement model $\Pr(\psi'_i|\psi)$ can be seen as the proportion of the belief that is carried from ψ to ψ'_i. This probability can be defined by the following models:

Empirical model: we count $n(\psi)$ from the training datasets, which is the number of user goals that are consistent with partition ψ. The probability is then modeled as $\Pr(\psi'_i|\psi) = \frac{n(\psi'_i)}{n(\psi)}$ if $n(\psi) > 0$ and $\Pr(\psi'_i|\psi) = 0$ otherwise.

Word-match model: this model extends the empirical model by using the domain knowledge when the SLU result $\tilde{u}_i$ does not appear in the training datasets. We calculated how many words $w \in W$ in the user utterance $\tilde{u}_i$ were included in a bus timetable $\mathcal{D}$. The model is thus defined as $\Pr(\psi'_i|\psi) = \frac{n(\psi'_i)}{n(\psi)}$ if $n(\psi'_i) > 0$ and $\Pr(\psi'_i|\psi) = \frac{\alpha}{|W|}\sum_{w \in W} \delta(w \in \mathcal{D})$ otherwise. δ is the indicator function ($\delta(x) = 1$ if x holds and $\delta(x) = 0$ otherwise) and α is the parameter estimated via cross-validation.

Mixture model: this model mixes the empirical model with a uniform probability, defined as $\Pr(\psi'_i|\psi) = \epsilon\frac{1}{n_G} + (1-\epsilon)\frac{n(\psi'_i)}{n(\psi)}$ if $n(\psi'_i) > 0$ and $\Pr(\psi'_i|\psi) = \frac{1}{n_G}$ otherwise. n_G is the number of all possible user goals which is treated as the parameter of the model and found via cross-validation, together with the mixing parameter $\epsilon \in [0, 1]$.

2.4 Belief Transformation Model

The belief update described above produces the M-best hypotheses of user goals $[\langle\tilde{g}_1, b(\tilde{g}_1)\rangle, \ldots, \langle\tilde{g}_M, b(\tilde{g}_M)\rangle]$ in each dialog turn, which consists of M most likely user goal hypotheses $\tilde{g}_i$ and their associated beliefs $b(\tilde{g}_i)$. The last hypothesis $\tilde{g}_M$ is reserved as the null hypothesis $\varnothing$ with the belief $b(\varnothing) = 1 - \sum_{i=1}^{M-1} b(\tilde{g}_i)$, which represents that the user goal is not known up to the current dialog turn.

One of the problems with the belief update is that the null hypothesis often remains as the most probable hypothesis even when the SLU result contains the correct user utterance with a high confidence score. This is because an atomic hypothesis has a very small prior probability.

To overcome this problem, we added a post-processing step which transforms each belief $b(h_i)$ to the final confidence score s_i.

Threshold model: this model ensures that the top hypothesis has confidence score θ when a belief of the hypothesis is greater than a threshold δ. The final output list is $[\langle h^*, s^*\rangle, \langle\varnothing, 1-s^*\rangle]$ where $h^* = \text{argmax}_{h \in \{\tilde{g}_1,\ldots,\tilde{g}_{M-1}\}} b(h)$ and

$$s^* = \begin{cases} \theta, & \text{if } b(h^*) > \delta \\ b(h^*), & \text{otherwise.} \end{cases} \quad (2)$$

Full-list regression model: this model estimates the probability that each hypothesis is correct via casting the task as regression. The model uses two logistic regression functions $F_\varnothing$ and F_h. $F_\varnothing$ predicts the probability of correctness for the null hypothesis $\varnothing$ using the single input feature $\phi_\varnothing = b(\varnothing)$. Likewise, F_h predicts the probability of correctness for non-null hypotheses h_i using the input feature $\phi_i = b(h_i)$. The model generates the final output list $[\langle h_1, s_1\rangle, \ldots, \langle h_{M-1}, s_{M-1}\rangle, \langle\varnothing, s_M\rangle]$ where $h_i = \tilde{g}_i$ and

$$s_i = \begin{cases} \frac{F_\varnothing(\phi_i)}{\sum_{j=1}^{M-1} F_h(\phi_j) + F_\varnothing(\phi_\varnothing)}, & \text{if } i = M \\ \frac{F_h(\phi_i)}{\sum_{j=1}^{M-1} F_h(\phi_j) + F_\varnothing(\phi_\varnothing)}, & \text{otherwise.} \end{cases} \quad (3)$$

Rank regression model: this model works in a similar way as in the full-link regression model, except that it uses a single logistic regression function F_r for both the non-null and null hypotheses, and takes the rank value of the hypotheses as an additional input feature. The final output list is $[\langle h_1, s_1\rangle, \ldots, \langle h_{M-1}, s_{M-1}\rangle, \langle \varnothing, s_M\rangle]$ where $h_i = \tilde{g}_i$ and

$$s_i = \frac{F_r(\phi_i)}{\sum_{j=1}^{M} F_r(\phi_j)}. \tag{4}$$

3 Experimental Setup

In the experiments, we used three labeled training datasets (train1a, train2, train3) and three test datasets (test1, test2, test3) used in DSTC. There was an additional test dataset (test4), which we decided not to include in the experiments since we found that a significant number of labels were missing or incorrect.

We measured the tracker performance according to the following evaluation metrics used in DSTC[1]: **accuracy (acc)** measures the rate of the most likely hypothesis h_1 being correct, **average score (avgp)** measures the average of scores assigned to the correct hypotheses, **l2 norm** measures the Euclidean distance between the vector of scores from the tracker and the binary vector with 1 in the position of the correct hypotheses, and 0 elsewhere, **mean reciprocal rank (mrr)** measures the average of $1/R$, where R is the minimum rank of the correct hypothesis, **ROC equal error rate (eer)** is the sum of false accept (FA) and false reject (FR) rates when FA rate=FR rate, and **ROC.**$\{$**v1,v2**$\}$**.**P measures correct accept (CA) rate when there are at most $P\%$ false accept (FA) rate[2].

4 Results and Analyses

Since there are multiple slots to track in the dialog domain, we report the average performance over the "marginal" slots including the "joint" slot that assigns the values to all slots.

4.1 Observation Model

Tbl. 1 shows the cross-validation results of the three observation models. In train1a and train2, no model had a clear advantage to others, whereas in train3, the confidence score model outperformed others. Further analyses revealed that the confidence scores from the SLU results were not sufficiently indicative of the SLU accuracy in train1a and train2. The histogram and the generative models are expected to perform at least as well as the confidence score model in train3, but they didn't in the experiments. We suspect that this is due to the naive binning strategy we used to model the probability distribution.

Table 1: Evaluation of observation models.

	Train1a			Train2			Train3		
	Conf	Hist	Gen	Conf	Hist	Gen	Conf	Hist	Gen
accuracy	0.81	0.82	**0.82**	0.84	**0.86**	0.85	**0.90**	0.89	0.88
avgp	0.77	0.78	**0.78**	0.81	**0.82**	0.82	**0.81**	0.79	0.77
l2	0.31	0.30	**0.30**	0.26	**0.25**	0.25	**0.25**	0.27	0.30
mrr	0.87	0.87	**0.88**	0.89	**0.89**	0.89	**0.94**	0.93	0.92
roc.v1.05	0.69	**0.70**	0.70	0.73	0.74	**0.74**	**0.82**	0.80	0.79
roc.v1.10	0.74	0.75	**0.75**	0.78	0.80	**0.80**	**0.87**	0.85	0.83
roc.v1.20	0.78	0.79	**0.79**	0.83	0.84	**0.84**	**0.89**	0.87	0.85
roc.v1.eer	0.14	**0.14**	0.14	**0.12**	0.13	0.13	**0.10**	0.11	0.12
roc.v2.05	**0.34**	0.34	0.34	**0.24**	0.15	0.23	0.52	**0.54**	0.52
roc.v2.10	**0.54**	0.46	0.46	**0.33**	0.26	0.25	**0.71**	0.67	0.70
roc.v2.20	**0.70**	0.70	0.69	**0.43**	0.41	0.41	**0.83**	0.78	0.80

Table 2: Evaluation of belief refinement models.

	Train1a			Train2			Train3		
	Emp	Word	Mix	Emp	Word	Mix	Emp	Word	Mix
accuracy	0.75	0.77	**0.81**	0.80	0.84	**0.84**	0.71	0.88	**0.90**
avgp	0.75	0.76	**0.77**	0.78	0.80	**0.81**	0.68	0.80	**0.81**
l2	0.34	0.34	**0.31**	0.31	0.27	**0.26**	0.42	0.26	**0.25**
mrr	0.83	0.85	**0.87**	0.86	0.89	**0.89**	0.82	0.93	**0.94**
roc.v1.05	0.66	0.68	**0.69**	0.64	0.68	**0.73**	0.58	0.78	**0.82**
roc.v1.10	0.69	0.71	**0.74**	0.73	0.78	**0.78**	0.65	0.83	**0.87**
roc.v1.20	0.73	0.74	**0.78**	0.77	0.82	**0.83**	0.68	0.86	**0.89**
roc.v1.eer	0.22	**0.13**	0.14	0.13	0.13	**0.12**	0.13	0.11	**0.10**
roc.v2.05	0.34	0.24	**0.34**	**0.30**	0.24	0.24	**0.61**	0.51	0.52
roc.v2.10	0.47	0.38	**0.54**	**0.42**	0.26	0.33	0.64	0.67	**0.71**
roc.v2.20	**0.72**	0.60	0.70	**0.56**	0.37	0.43	0.72	0.77	**0.83**

4.2 Belief Refinement Model

As shown in Tbl. 2, the mixture model outperformed others throughout the metrics. It even outperforms the word-match model which tries to leverage the domain knowledge to handle novel user goals. This implies that, unless the domain knowledge is used properly, simply taking the mixture with the uniform distribution yields a sufficient level of performance.

4.3 Belief Transformation Model

Tbl. 3 summarizes the performances of the belief transformation models. All three models outperformed the pure belief update, although not shown

[1] http://research.microsoft.com/apps/pubs/?id=169024

[2] There are two types of ROC measured in DSTC depending on how CA and FA rates are calculated. The detailed discussion is provided in the longer version of the paper (Kim et al., 2013).

Table 3: Evaluation of belief transform models.

	Train1a			Train2			Train3		
	Thre	Full	Rank	Thre	Full	Rank	Thre	Full	Rank
accuracy	0.81	**0.81**	0.81	0.83	0.84	**0.85**	0.89	**0.90**	0.90
avgp	**0.80**	0.77	0.77	**0.82**	0.81	0.81	**0.85**	0.81	0.78
l2	**0.28**	0.31	0.32	**0.25**	0.26	0.26	**0.22**	0.25	0.28
mrr	0.84	**0.87**	0.87	0.86	0.89	**0.89**	0.91	**0.94**	0.92
roc.v1.05	0.66	0.69	**0.69**	0.65	**0.73**	0.72	0.45	**0.82**	0.80
roc.v1.10	0.71	0.74	**0.75**	0.69	0.78	**0.79**	0.68	**0.87**	0.86
roc.v1.20	0.71	0.78	**0.78**	0.74	0.83	**0.83**	0.79	**0.89**	0.89
roc.v1.eer	0.18	0.14	**0.14**	0.21	0.12	**0.12**	0.49	0.10	**0.09**
roc.v2.05	0.22	**0.34**	0.34	0.20	0.24	**0.24**	0.42	**0.52**	0.48
roc.v2.10	0.41	**0.54**	0.52	0.22	**0.33**	0.33	0.42	**0.71**	0.56
roc.v2.20	0.64	0.70	**0.71**	0.30	0.43	**0.49**	0.43	**0.83**	0.75

in the table. The full-list and the rank regression models show a similar level of performance improvement. This is a naturally expected result since they use regression to convert the beliefs to final confidence scores, as an attempt to compensate for the errors incurred by approximations and assumptions made in the observation and belief refinement models.

4.4 DSTC Result

In order to compare our tracker with others participated in DSTC, we chose tracker4[3] as the most effective one among our 5 submitted trackers since it achieved the top scores in the largest number of evaluation metrics. In the same way, we selected tracker2 for team3, tracker3 for team6, tracker3 for team8, and tracker1 for the rest of the teams. The results of each team are presented in Tbl. 4. The baseline tracker is included as a reference, which simply outputs the hypothesis with the largest SLU confidence score in the N-best list.

Compared to other teams, our tracker showed strong performance in acc, avgp, l2 and mrr. A detailed discussion on the results is provided in the longer version of the paper (Kim et al., 2013).

5 Conclusion

In this paper, we described our experience with engineering a statistical dialog state tracker while participating in DSTC. Our engineering effort was focused on improving three important models in the tracker: the observation, the belief refinement, and the belief transformation models. Using standard statistical techniques, we were able

[3]The tracker4 used the confidence score model, the mixture model and the rank regression model.

Table 4: Results of the trackers. The bold face denotes top 3 scores in each evaluation metric. T9 is our tracker.

	Base	T1	T2	T3	T4	T5	T6	T7	T8	T9
Test 1										
accuracy	0.71	**0.83**	0.81	0.81	0.74	0.80	**0.87**	0.78	0.51	**0.82**
avgp	0.73	0.77	0.77	**0.81**	0.74	0.79	**0.82**	0.76	0.49	**0.79**
l2	0.38	0.32	0.32	**0.27**	0.37	0.30	**0.25**	0.34	0.72	**0.29**
mrr	0.80	**0.88**	0.86	0.85	0.81	0.85	**0.90**	0.84	0.59	**0.88**
roc.v1.05	0.62	**0.72**	0.67	0.60	0.20	0.71	**0.76**	0.65	0.20	**0.72**
roc.v1.10	0.63	**0.78**	0.75	**0.77**	0.29	0.75	**0.82**	0.70	0.33	0.76
roc.v1.20	0.67	**0.82**	0.79	0.79	0.53	0.78	**0.85**	0.76	0.35	**0.79**
roc.v1.eer	0.24	**0.13**	0.25	0.24	0.74	**0.12**	**0.12**	0.15	0.52	0.14
roc.v2.05	**0.49**	**0.64**	0.01	0.02	0.00	**0.55**	0.16	0.19	0.04	0.26
roc.v2.10	**0.69**	**0.71**	0.14	0.03	0.00	**0.68**	0.39	0.35	0.05	0.47
roc.v2.20	**0.71**	**0.80**	0.48	0.29	0.00	**0.74**	0.59	0.58	0.27	0.62
Test 2										
accuracy	0.55	0.65	**0.71**	0.68	0.63	0.62	**0.79**	0.65	0.34	**0.71**
avgp	0.57	0.55	0.63	**0.68**	0.63	0.62	**0.71**	0.65	0.29	**0.65**
l2	0.60	0.63	0.50	**0.45**	0.52	0.54	**0.39**	0.49	1.00	**0.48**
mrr	0.65	0.72	**0.79**	0.76	0.71	0.72	**0.84**	0.74	0.46	**0.80**
roc.v1.05	0.43	0.49	**0.52**	0.45	0.16	0.48	**0.66**	0.48	0.04	**0.49**
roc.v1.10	0.45	0.54	**0.57**	**0.63**	0.16	0.51	**0.71**	0.54	0.11	0.57
roc.v1.20	0.48	0.59	**0.64**	**0.64**	0.27	0.54	**0.76**	0.60	0.26	0.63
roc.v1.eer	0.19	0.20	0.39	**0.14**	0.63	0.21	**0.16**	**0.19**	0.36	0.22
roc.v2.05	**0.43**	**0.52**	0.24	0.27	0.00	0.40	**0.46**	0.41	0.05	0.38
roc.v2.10	0.47	**0.60**	0.40	0.37	0.00	**0.62**	**0.53**	0.47	0.17	0.41
roc.v2.20	0.50	**0.70**	0.48	0.56	0.00	**0.70**	**0.62**	0.55	0.44	0.47
Test 3										
accuracy	0.79	0.79	**0.84**	0.82	0.82	0.78	**0.84**	0.79	0.79	**0.85**
avgp	0.75	0.72	0.76	**0.79**	**0.78**	0.70	0.75	0.75	**0.76**	0.74
l2	0.35	0.37	0.32	**0.29**	**0.30**	0.40	0.33	0.34	**0.32**	0.34
mrr	0.83	0.85	**0.88**	0.85	0.85	0.83	**0.89**	0.84	0.80	**0.89**
roc.v1.05	0.56	0.65	0.68	**0.72**	**0.70**	0.62	0.69	0.70	0.33	**0.74**
roc.v1.10	0.66	0.70	**0.77**	**0.77**	0.76	0.69	0.76	0.74	0.47	**0.78**
roc.v1.20	0.74	0.76	**0.82**	0.80	0.80	0.74	**0.81**	0.77	0.61	**0.82**
roc.v1.eer	0.19	0.16	0.15	0.27	**0.12**	0.17	0.15	**0.12**	0.34	**0.13**
roc.v2.05	0.56	**0.62**	0.34	0.28	0.21	**0.62**	**0.61**	0.14	0.00	0.56
roc.v2.10	0.59	**0.71**	0.48	0.37	0.52	0.66	**0.66**	0.42	0.00	**0.67**
roc.v2.20	0.66	0.78	0.73	0.52	**0.82**	0.71	0.78	**0.87**	0.00	**0.79**

to produce a tracker that performed competitively among the participants.

As for the future work, we plan to refine the user utterance model for improving the performance of the tracker since there are a number of user utterances that are not handled by the current model. We also plan to re-evaluate our tracker with properly handling the joint slot, since the current tracker constructs models independently for each marginal slot and then combines the results by simply multiplying the predicted scores.

Acknowledgement

This work was supported by NRF of Korea (Grant# 2012-007881), and MKE/KEIT of Korea (IT R&D Program Contract# 10041678)

References

Dan Bohus and Alex Rudnicky. 2006. A "k hypotheses + other" belief updating model. In *Proceedings of the AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.

Dongho Kim, Jin Hyung Kim, and Kee-Eung Kim. 2011. Robust performance evaluation of POMDP-based dialogue systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):1029–1040.

Daejoong Kim, Jaedeug Choi, Kee-Eung Kim, Jungsu Lee, and Jinho Sohn. 2013. Engineering statistical dialog state trackers:a case study on DSTC. Technical Report CS-TR-2013-379, Department of Computer Science, KAIST.

Neville Mehta, Rakesh Gupta, Antoine Raux, Deepak Ramachandran, and Stefan Krawczyk. 2010. Probabilistic ontology trees for belief tracking in dialog systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 37–46.

Nicholas Roy, Joelle Pineau, and Sebastian Thrun. 2000. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 93–100.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

Jason D. Williams. 2008. Exploiting the ASR N-best by tracking multiple dialog state hypotheses. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 191–194.

Jason D. Williams. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 5382–5385.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.