Challenges and opportunities for state tracking in statistical spoken dialog systems: results from two public deployments

Jason D. Williams, *Member, IEEE*

***Abstract*—Whereas traditional dialog systems operate on the top ASR hypothesis, statistical dialog systems claim to be more robust to ASR errors by maintaining a distribution over multiple hidden dialog states. Recently, these techniques have been deployed publicly for the first time, making empirical measurements possible. In this paper, we analyze two of these deployments. We find that performance was quite mixed: in some cases statistical techniques improved accuracy with respect to the top speech recognition hypothesis; in other cases, accuracy was degraded. Investigating degradations, we find the three main causes are (non-obviously) inaccurate parameter estimates, poor confidence scores, and correlations in speech recognition errors. Overall the results suggest fundamental weaknesses in the formulation as a generative model, and we suggest alternatives as future work.**

***Index Terms*—Spoken dialog systems, human-computer interaction.**

I. Introduction

For more than a decade, researchers have worked to apply statistical techniques to spoken dialog systems [1], [2], [3], [4], [5]. By learning from data and experience, these techniques seek to outperform manually designed systems. Two broad problems have been studied. The first problem is *accurately tracking the state of the dialog*. This problem is difficult in large part because the speech recognition and language understanding processes are error-prone, making the true state of the dialog only partially observable. Here one of the main aims of statistical techniques is to improve robustness to recognition errors. The second problem is *choosing system actions*. Dialog is a temporal process, so actions must be chosen to satisfy long-term goals. This is particularly challenging given the uncertainty in the state of the dialog, and here the main aims are creating dialog plans that are optimal with respect to some criteria.

This paper is concerned with the first problem, *accurately tracking the state of the dialog*. A popular approach is to maintain a distribution over many possible hypotheses for the true state of the dialog, by globally synthesizing all of the observable history with statistical models estimated from training dialogs. These methods were initially developed on toy problems [4], [3], [5], then tested in simulation [6], [7] and controlled laboratory studies [8], [9], [10], [11]. Recently, in the Spoken Dialog Challenge [12], the first public deployments have been done, providing the first opportunity to empirically assess real-world performance.

TABLE I: Two dialog systems studied in this paper. Utterance counts for each slot show the number of non-empty utterances received in response to system requests for that slot.

	DS1	DS2
Active	Summer 2010	Winter 2011-2
Calls	779	1037
Utterances	9636	13484
Mean utts/call	12.4	13.0
route utterances	1495	2955
from utterances	1197	1656
to utterances	1148	1592
day utterances	175	128
time utterances	155	237
TOTAL	4170	6568

This paper considers two versions of one of the first statistical dialog systems deployed to the public. The first version (DS1) was deployed in 2010 [13], and the second version (DS2) was deployed in 2011-2012. This paper presents a detailed analysis of these deployments, with a particular emphasis on the causes of state tracking errors in statistical spoken dialog systems. The contribution of this paper is not a new technique or algorithm, but rather a thorough evaluation of state-of-the-art technology in real-world use, with the aim of informing future research efforts.

Existing work studied DS1 in some detail [14]. Issues that were found in DS1 were addressed in the creation of DS2. This paper subsumes this past work [14], presenting a joint analysis of both systems. New insights in this paper include: empirical data showing the relationship between accuracy and the quality of model parameters; identification of correlations in speech recognition errors as a major cause of failures; and evidence for fundamental flaws in several components of current models. Taken together, these findings suggest that discriminative (rather than generative) approaches may be promising for belief tracking.

In this paper, Section II reviews algorithms for statistical spoken dialog systems. Section III then describes the two dialog systems under study. Section IV reports on overall accuracy, then analyzes the underlying reasons for accuracy gains and losses. Section V tackles how well errors can be identified, and Section VI concludes by summarizing lessons learned and suggesting new avenues for research.

J. D. Williams is with Microsoft Research, Redmond, WA, USA. jason.williams@microsoft.com

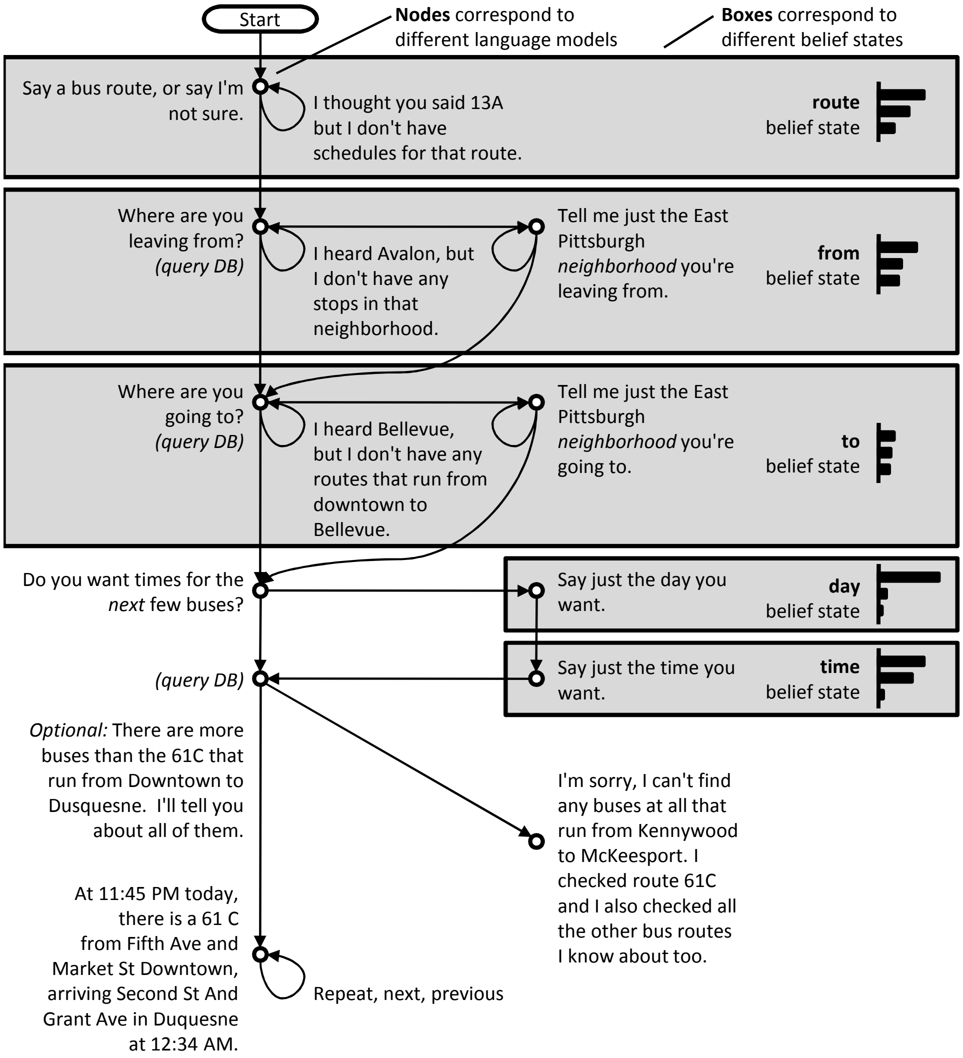

Start
Nodes correspond to different language models
Boxes correspond to different belief states
Say a bus route, or say I'm not sure.
I thought you said 13A but I don't have schedules for that route.
route belief state
Where are you leaving from? *(query DB)*
I heard Avalon, but I don't have any stops in that neighborhood.
Tell me just the East Pittsburgh *neighborhood* you're leaving from.
from belief state
Where are you going to? *(query DB)*
I heard Bellevue, but I don't have any routes that run from downtown to Bellevue.
Tell me just the East Pittsburgh *neighborhood* you're going to.
to belief state
Do you want times for the *next* few buses?
Say just the day you want.
day belief state
(query DB)
Say just the time you want.
time belief state
Optional: There are more buses than the 61C that run from Downtown to Dusquesne. I'll tell you about all of them.
I'm sorry, I can't find any buses at all that run from Kennywood to McKeesport. I checked route 61C and I also checked all the other bus routes I know about too.
At 11:45 PM today, there is a 61 C from Fifth Ave and Market St Downtown, arriving Second St And Grant Ave in Duquesne at 12:34 AM.
Repeat, next, previous

Fig. 1: Hand-crafted design followed by DS1 and DS2. The system asks for the bus route, then the origin bus stop from, then the destination bus stop to. If the user does not want the next few buses, the system also asks for the day and time. Prompts shown are paraphrases; actual system prompts include example responses and are tailored to dialog context. Different language models are used for each slot, and separate belief states are maintained over each of these 5 slots

II. State tracking in statistical dialog systems

Statistical dialog systems maintain a distribution over a set of hidden dialog states. A dialog state includes information not directly observable to the dialog system, typically the user's overall goal in the dialog, or the user's true action (e.g., the user's true dialog act). For example, at turn t in a bus timetable system, the user's overall goal might be to go from downtown Pittsburgh to Carnegie Mellon University on the next bus, and the user's action might be to say "leaving from downtown".

For each dialog state s, a posterior probability of correctness called a *belief* is maintained $b(s)$. The set of hidden dialog states and their beliefs is collectively called the *belief state*, and updating the belief state is called *belief tracking*.

At the start of the dialog, the belief state is initialized to a *prior* distribution $b_0(s)$. The system then takes an action a, and the user takes an action in response. The automatic speech recognizer and spoken language understanding – collectively called "ASR" in this paper – then produces a ranked list of N hypotheses for the user's action, $\mathbf{u} = (u_1, \ldots, u_N)$, called an *N-best list*. For each N-best list, $P_{\text{asr}}(u)$ assigns a local, history-independent probability of correctness to each item, often called a *confidence score*. The belief state is then updated:

$$b'(s) = k \cdot \sum_u P_{\text{asr}}(u) P_{\text{act}}(u|s,a) b(s) \qquad (1)$$

where $P_{\text{act}}(u|s,a)$ is the probability of the user taking action u given the dialog is in hidden state s and the system performed action a. k is a normalizing constant. A full derivation of Eq 1 is given in Appendix A.

In words, first each item in the ASR N-Best list is assigned a local probability of correctness $P_{\text{asr}}(u)$. This confidence score indicates the local probability that the user took action u *independent of* the current dialog state. Then, the new belief in state s is given by multiplying factors for the ASR confidence score, the probability of the user taking the action, and the previous belief in the state.

TABLE II: Example labels.

User speech	ASR result	Label	Notes
forbes and murray	forbes and murray	correct	Lexically identical
hazelwood at emahlea	hazelwood and emahlea	correct	Refers to same intersection
ingram bus station	ingram station	correct	Refers to same station
braddock pennyslvania	braddock	correct	Refers to same location
braddock north braddock	braddock	incorrect	Incomplete
uh beechwood boulevard and murray avenue	beechwood boulevard	incorrect	Incomplete
forbes downtown	downtown	incorrect	Incomplete
hawkins village on south braddock avenue	hawkins village in rankin	incorrect	Contains wrong information
braddock pennyslvania	braddock avenue	incorrect	Street vs. neighborhood
arlington and brownsville	arlington and freeland	incorrect	Different intersection

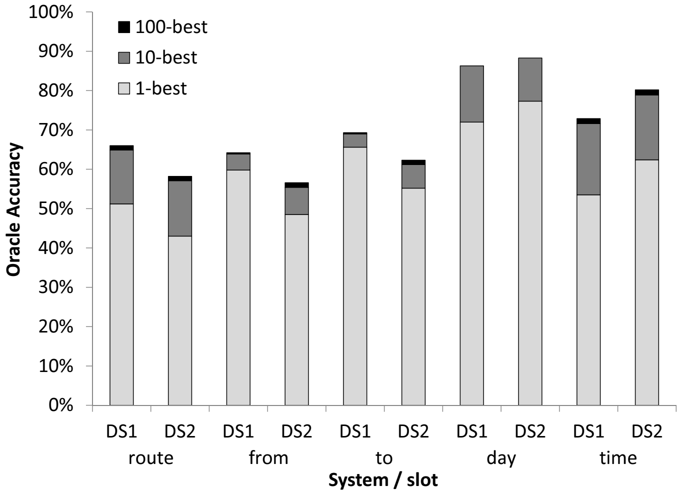


Fig. 2: ASR accuracy for 1-best, 10-best, and 100-best ASR N-best lists.

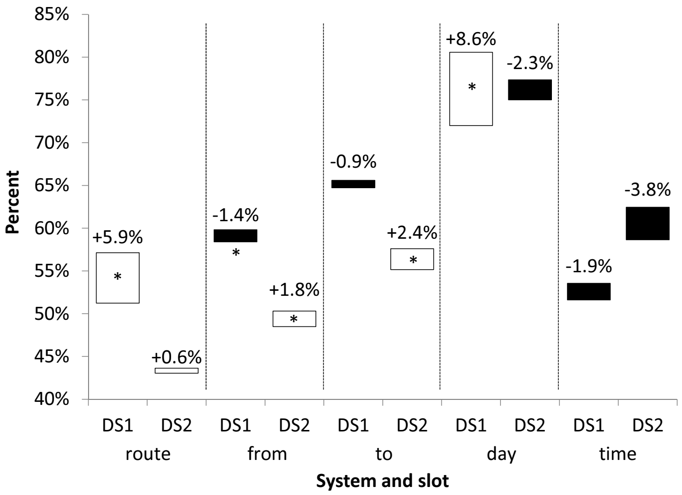


Fig. 3: Summary of accuracy. The tops and bottoms of each bar show accuracy for s^* and u_1. Unshaded bars indicate that the accuracy of s^* is higher than u_1 (ie, s^* corresponds to the top of the bar, and u_1 to the bottom). Shaded bars indicate that the accuracy of s^* is lower than u_1 (ie, s^* corresponds to the bottom of the bar, and u_1 to the top). Asterisk (*) indicates the difference is statistically significant with $p \leq 0.05$ using McNamara's Test.

In practice specialized techniques must be used to compute Eq 1 in real-time. The systems in this paper use *incremental partition recombination* [7]; alternative algorithms include the Hidden Information State [10], [15], Bayesian Update of Dialog States [11], distributions over frames [16], probabilistic ontology trees [17], and particle filters [18]. The details are not important for this paper – the key idea is that Eq 1 synthesizes a prior distribution over dialog states together with all of the ASR N-best lists and local confidence scores to form a cumulative, whole-dialog posterior probability distribution over all possible dialog states, $b(s)$.

(or a blank line to force the subfigure onto a new line)

The claimed benefit is that – provided the models are estimated well – the dialog state with the highest belief $s^* = \arg\max_s b(s)$ should be correct more often than a dialog state constructed heuristically from the 1-best ASR results. This is the main claim we evaluate in this paper.

III. Dialog systems under study

The two systems under study in this paper – DS1 and DS2 – provide bus timetable information for Pittsburgh, USA. They were fielded to the public as a part of the Spoken Dialog Challenge [19], [12]. The systems themselves were fielded by AT&T [13], and the analysis in this paper is based on audio and logs from these systems, publicly available from the Dialog Research Center at Carnegie Mellon University [20].

As with most commercial dialog systems, they followed a highly directed flow, collecting one *slot* at a time. There are five slots: route, from, to, day, and time. These systems could only recognize values for the slot being queried, plus a handful of global commands ("repeat", "go back", "start over", "goodbye", etc.) – mixed initiative and over-completion were not supported.

The common design of the systems is shown in Figure 1. Each system opened by asking the user to say a bus route, or to say "I'm not sure." The systems could recognize any of the $\sim$100 routes in Pittsburgh, but could only provide times for a *covered* subset of routes. If an uncovered route was recognized, the system explained that it only had information for certain routes. Otherwise, the system next asked for the from and to slots. If a bus route was specified, the language models for from and to include only the locations along that

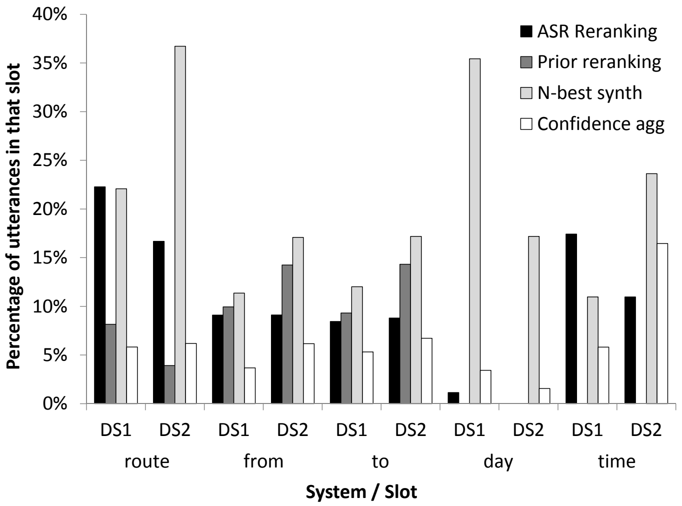


Fig. 4: Frequency of occurrence of each mechanism on each slot

route, including streets, intersections, and landmarks, reducing the complexity of the location recognition task. If repeated non-understandings or mis-understandings were detected, the system backed off to asking for neighborhoods instead.

After gathering the bus route and locations, the system then asked if the caller wants times for the "next few buses". If not, the system asked for the day then time in two separate questions. Finally bus times were read out. Users could say "start over" at any time.

Belief tracking was done with the AT&T Statistical Dialog Toolkit [21], and an independent belief state was maintained for each slot. After requesting the value of a slot, the system received an ASR N-best list, assigned each item a confidence score $P_{\text{asr}}(u)$, and updated the belief in (only) that slot using Eq 1. The top dialog hypothesis $s^* = \arg\max_s b(s)$ and its belief $b(s^*)$ were used to determine which action to take next, following a hand-crafted policy. This is in contrast to conventional dialog systems, in which the top ASR result and its confidence score govern dialog flow. If the belief was high, the slot was implicitly confirmed ("Ok, route 61C. To change, say go back. Where are you leaving from?"); if the belief was medium, the slot was explicitly confirmed ("I heard 61C. Is that right?"); if the belief was low, the question was asked again ("Sorry, say a bus route, or say I'm not sure."). The thresholds defining high, medium, and low were set by hand and were identical for all slots.[1]

Confidence scores $P_{\text{asr}}(u)$ were assigned using a two-stage model [22]. In the first stage, a maximum entropy classifier assigned a probability to three classes, where the classes indicate (1) that the top ASR result u_1 is correct; (2) that one of the items in $u_2 \ldots u_N$ is correct; and (3) that none of the items on the ASR N-best list is correct. In the second stage, a Beta distribution is used to allocate the probability of class (2) across items $u_2 \ldots u_N$. The maximum entropy classifier and Beta distribution were trained on data (details in Section IV-A). Note that the structure of the confidence score model $P_{\text{asr}}(u)$ made it possible for item $n = 2$ to be assigned a higher confidence score than $n = 1$, although this wasn't necessarily desired.

The two systems were identical, except for the following:

- DS1 could provide timetables for 8 routes; DS2 could provide timetables for ∼40 routes. To suggest the scope of functionality, DS1 opened with "East Pittsburgh bus times"; DS2 opened with "Pittsburgh bus times".
- The names and times of some bus routes in Pittsburgh changed between DS1 and DS2
- DS2 used a different acoustic model than DS1
- The systems used different voices for synthesized speech and recorded prompts
- DS2 used different priors b_0 than DS1
- DS2 used different training data to estimate $P_{\text{asr}}(u)$

The last two changes were made in an attempt to improve shortcomings discovered in DS1 [14], and will be discussed more below.

DS1 received 779 calls in the period July 16 – August 16 2010, containing a total of 9,636 user utterances, of which 4,170 contained non-empty responses to requests for one of the five slots. The remainder were responses to yes/no questions, timetable navigation commands like "next bus", silence, etc. DS2 received 1,037 calls from 28 December 2011 – 6 February 2012, containing a total of 13,484 utterances, of which 6,605 contained non-empty responses to requests for one of the five slots. Table I provides details. There are relatively fewer date and time utterances because most callers asked for the next few buses, in which case the caller was not asked for the date and time.

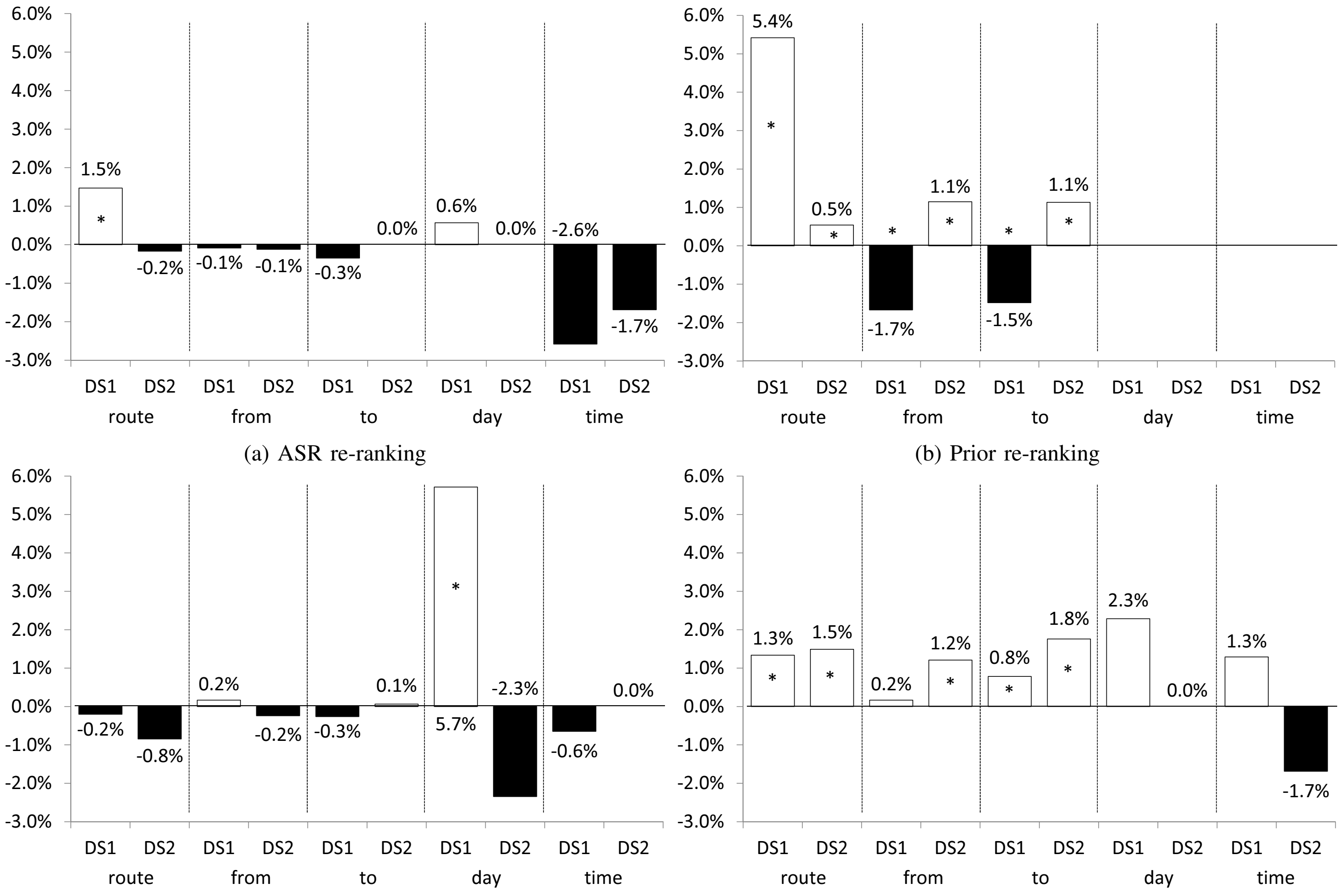
IV. Analysis of accuracy

As explained above, in the system in this paper, slots are queried separately, and an independent belief state is maintained for each. Consequently, within each slot user actions u and hidden states s are drawn from the same set of slot values. Thus, to measure the performance within each slot, we will compare the accuracy of the top belief state $s^* = \arg\max_s b(s)$ to the accuracy of the top ASR result u_1 (our baseline). Since we are interested in task performance, throughout the paper we'll use semantic accuracy, not word accuracy.

We began by selecting utterances containing non-empty responses to each of the five slots (counts in Table I). A professional transcriber (not the author) listened to each utterance. Following a labeling guide, they marked each hypothesis on the ASR N-best list as *correct* if it was *semantically* consistent with the user's speech, or *incorrect* otherwise. These labels were then checked by a second professional transcriber. Example labels and excerpts from the labeling guide are shown in Table II.

Basic ASR accuracy results are shown in Figure 2. 1-best ASR accuracy for route, from, and to was lower in DS2 than

[1] It is possible that performance gains could be achieved by tuning the thresholds, and in commercial systems ASR confidence score thresholds are often tuned to maximize performance. However the thresholds were not tuned for two reasons. First, practically, the systems were operational for relatively short timescales, making it difficult to complete the transcription feedback loop. Second, in principle, with statistical methods the belief corresponds to a proper probability, and this ought to allow thresholds to be specified without tuning.



(a) ASR re-ranking

(b) Prior re-ranking

(c) N-best synthesis

(d) Confidence aggregation

Fig. 5: Effects of each mechanism on each slot. Each bar shows $(x - y)/z$, where x is the number of utterances where the mechanism occurred *and* the belief 1-best is correct, y is the number of utterances where the mechanism occurred *and* the ASR 1-best is correct, and z is the total number of utterances in that slot/system (regardless of whether the mechanism occurred). Asterisk (*) indicates the difference is statistically significant with $p \leq 0.05$ using McNamara's Test.

DS1, which is unsurprising since DS2 covered more routes and locations than DS1. day and time used identical language models in DS1 and DS2; 1-best ASR accuracy was slightly higher for these slots in DS2, which may reflect differences in the acoustic models between DS1 and DS2. Oracle accuracy was 5-15% larger, with most of the gain in the first 10 items on the ASR N-best list.

We next determined the accuracy of the top belief state s^*. In these systems, each item in the belief state maps directly to one or more ASR hypotheses. In addition, typically the user's goal remains fixed throughout the call, at least until the caller says "start over". Given this, the correctness of the top belief state was set to the correctness of the most recent ASR hypothesis it mapped to. However, if the user said "start over", the set of relevant ASR items was cleared.

For example, at turn t, the top item in the belief state s^* might map to the third item in the ASR N-best list (u_3) in turn $t-1$, and to the second item in the ASR N-best list (u_2) in turn t. If u_2 in turn t were labeled as incorrect, the top belief state in turn t would be considered incorrect.

Results are in Figure 3, which shows the absolute difference in accuracy between the ASR 1-best u_1 and the belief state 1-best s^*. While belief tracking yielded an improvement in accuracy in some cases, it caused a degradation in others.

We next sought to understand the causes of this varied performance. Formally, differences between the top ASR result u_1 and the top belief state s^* are simply the result of evaluating Eq 1. However, *intuitively* there are four *mechanisms* which cause differences, and each difference can be explained by the action of one or more mechanisms. These mechanisms are summarized here; Appendix B provides graphical illustrations.[2]

- **ASR re-ranking**: Recall that our confidence score $P_{\text{asr}}(u)$ had the ability to assign a higher confidence score to u_2 than u_1; when this *ASR re-ranking* happens, this may cause s^* to differ from u_1 (Figure 11a).
- **Prior re-ranking**: Statistical techniques use a prior prob-

[2] This taxonomy was developed for belief tracking over a single slot using the type of model employed in this paper. For systems that track joint beliefs over multiple slots, or which use different models for belief tracking, the taxonomy may be different. For example, the distinction between N-best synthesis and confidence aggregation may not be important in other belief tracking models.

ability for each possible dialog state – in our system, each slot value – $b_0(s)$. If an item recognized lower-down on the N-best list has a high prior, it can obtain the most belief, causing s^* to differ from u_1 (Figure 11b).

- **N-best synthesis**: If an item appears in two N-best lists, but is not in the top ASR N-best position in the latter recognition, it may still obtain the highest belief, causing s^* to differ from u_1 (Figure 11c).
- **Confidence aggregation**: If the top belief state s^* has high belief, then subsequent low-confidence recognitions which do not contain s^* will not dislodge s^* from the top position, causing s^* to differ from u_1 (Figure 11d).

These definitions were encoded, and mechanism occurrences were detected automatically. Figure 4 shows occurrence frequencies. In general, N-best synthesis was most common, and confidence aggregation was least common. Prior re-ranking did not occur at all in day and time because those slots used a flat prior.

We next examined co-occurrences between mechanisms. Confidence aggregation always occurs on its own, since – unlike the other 3 mechanisms – it is only possible when the slot value of the top belief state is *not* observed on the current N-best list. The remaining 3 mechanisms occur in isolation about two-thirds of the time.

Figure 5 shows the improvement/degradation of each mechanism on each slot/system. Although there are some trends, there is no overall pattern. The next four sections examine each mechanism in detail.

A. ASR re-ranking

Recall that the models that assigned (local) confidence scores P_{asr} were not specifically designed to re-rank the ASR N-best list, but as an artifact of their two-stage design, it was possible for them to assign a higher confidence score to the $n = 2$ item than the $n = 1$ item. We call this re-ordering *ASR re-ranking*. Looking at ASR accuracy, we found that ASR re-ranking consistently degraded ASR accuracy, with particularly large degradations for time (-3.9% in DS1 and -3.4% in DS2) and route (-1.9% in DS1 and -2.3% in DS2). The ASR accuracy of day in DS1 was improved slightly, $+0.6\%$.

DS1 and DS2 used different confidence models P_{asr}. When DS1 was launched, there was no same-system data available, so a large corpus of data from a different dialog system was used to train the models [23]. This mismatch was one possible cause of the degradation for DS1, so P_{asr} for DS2 was trained on data from DS1. However, as mentioned above, ASR re-ranking also reduced ASR accuracy in DS2. This suggests that mis-matched training data is not the primary cause. Rather, it seems a more sophisticated model for P_{asr} is required – i.e., one which is explicitly aware of the order of items on the N-best list.

B. Prior re-ranking

Non-uniform priors were used in only route, from, and to. Figure 5b shows that prior re-ranking improved accuracy for route, substantially for DS1 and marginally for DS2. It also improved accuracy for from and to in DS2, but degraded accuracy for these slots in DS1. The explanations for these results lay in key differences between DS1 and DS2.

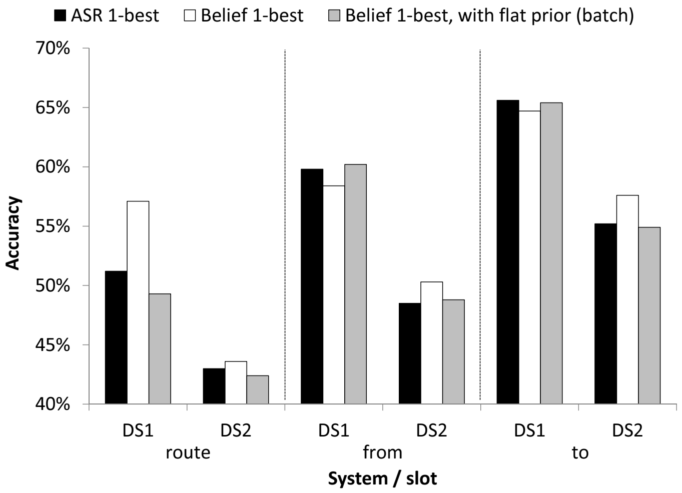


Fig. 6: Accuracy of ASR 1-best, belief 1-best, and belief 1-best re-scored with a flat (uniform) prior.

The first key difference between DS1 and DS2 is how priors were estimated. In DS1, an attempt was made to estimate priors using a heuristic that avoided collecting usage data. The heuristic assigned a prior proportional to the *number of bus stops* the slot value referred to. For example, for locations, "downtown" referred to many bus stops, but "the airport" referred to just one. In DS2, priors were estimated from actual usage observed in DS1.

For locations in DS1, this heuristic was a failure. The problem is that the heuristic did not reflect the fact that certain stops are more *popular* than others: for example, the airport corresponded to a single bus stop, but it was very popular; on the other hand, some neighborhoods had many bus stops but were almost never requested, perhaps because ridership in those neighborhoods was low. The net effect was that prior re-ranking for locations in DS1 degraded performance. In DS2, with priors estimated from (transcribed) usage data rather than a heuristic, priors yielded an improvement in accuracy for locations.

The second key difference between the systems is that the number of routes covered was much larger in DS2 than DS1. As mentioned above, the system could *recognize* any route, but could only provide times for *covered* routes – for others the system would report that it was unable to provide any information. Covered routes had high priors; others had very low priors. Most callers knew the set of covered routes, reinforced by the system opening "East Pittsburgh bus times" for DS1 and "Pittsburgh bus times" for DS2.

In DS1, the result was that most recognitions of non-covered routes were errors; the strong prior moved covered routes to the top of the belief state, yielding a large improvement for belief tracking for route in DS1. In DS2, a larger set of routes were now covered, so erroneous recognitions were no longer obvious; as a result, prior re-ranking still helped for route in DS2, but to a lesser extent.

To validate this explanation, belief tracking was re-run in

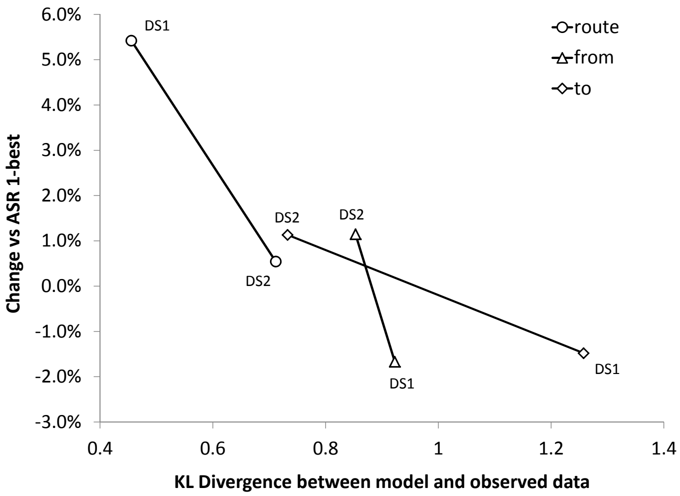


Fig. 7: Discrete KL divergence between model prior and observed data vs. the change in accuracy compared to the ASR 1-best. The y axis is computed as in Figure 5. Increasing KL divergence (i.e., poorer model fit) degrades the accuracy of belief tracking.

batch using a flat (uniform) prior. This removes the effect of the prior, so the difference between the flat prior and the ASR 1-best includes only the effects of the other 3 mechanisms. Results are in Figure 6. For from and to the flat prior result is very close to the ASR 1-best baseline. For route in DS1 and DS2, the flat prior result is worse than the ASR 1-best baseline: ASR re-ranking errors were highest for route, and without the prior to correct these errors, they reduce accuracy.

The overall trend appears to be that the effectiveness of prior re-ranking depends on how well the prior matches real use. To verify this, we plotted the discrete Kullback-Leibler (KL) divergence between the frequency of observation and the model for each of these 3 slots across the 2 systems. Figure 7 shows results. Within each slot, as the KL divergence increases, accuracy of belief tracking decreases.

C. N-best synthesis

Performance for N-best synthesis was quite varied. For route, from, to, and time, there was generally a negative (or marginal) effect. For day, there was a large improvement for DS1, and a moderate degradation for DS2.

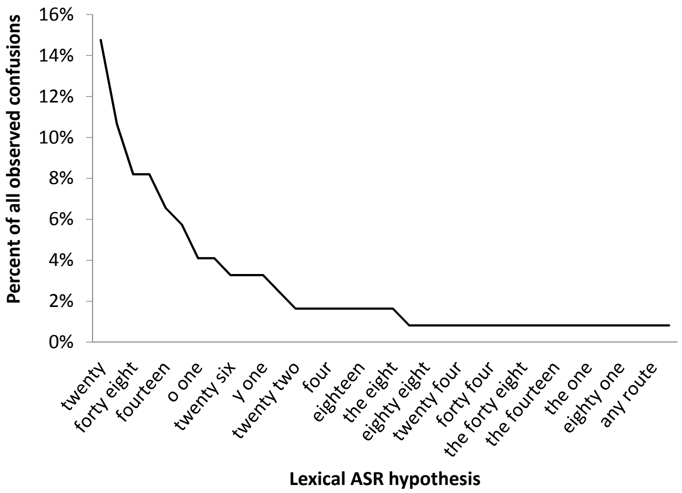


Fig. 8: Semantically incorrect items appearing in any location on the ASR N-best list when the user said *twenty eight x*. For space on the x axis, every second item is shown. The skew of the curve shows that confusions are highly correlated.

TABLE III: Example of how correlated ASR errors cause N-best synthesis to fail. The user says "sixty one a" twice. Items in bold are repeatedly mis-recognized. The correct item is underlined. Although the ASR 1-best is correct in the second turn, the belief state 1-best is incorrect due to repeated mis-recognitions of "sixty one".

Turn 1 System: which route? User: sixty one a		Turn 2 System: which route? User: sixty one a	
ASR	Belief state	ASR	Belief state
sixty one	**g one**	sixty one a	**sixty one**
g one	**sixty one**	**sixty one**	sixty one a
one	**the one; one**	the y one	**g one**
g	g	y one	**the one; one**
sixty	sixty	**one**	y one; the y one
p	p	**the one**	sixty one c
eleven	eleven	sixty one d	sixty one d
six	six	one d	g
fifty one	fifty one	the one d	sixty
the one	sixteen	sixty one c	p
sixteen		**g one**	eleven

Past work on DS1 suggested there were properties of N-best lists that govern the effect of N-best synthesis [14]; however, we did not see those trends in DS2. So, to understand the causes, each instance of a degradation caused by N-best synthesis was examined by hand. It was quickly evident that the primary cause of degradations was *correlated ASR errors*: i.e., the same recognition error occurring repeatedly. The key problem is that the update in Eq 1 – in particular P_{asr} – assumes that confusions are distributed uniformly: correlations cause repeated errors to be wrongly assigned too much belief mass. As an example, Figure 8 shows observed confusions – anywhere on the ASR N-best list – to the user saying the (exact) words "twenty eight x". A small number of items account for most of the confusions. Table III shows an excerpt from a real call that illustrates how these correlated ASR errors cause failures for N-best synthesis. Repeated confusions of "sixty one" for the user's request of "sixty one a" cause "sixty one" to erroneously obtain the highest belief.

Table IV shows that correlated ASR errors are to blame in 86% of erroneous instances of N-best synthesis. If these were eliminated, the overall effect of N-best synthesis would be positive.

Looking at day in DS2, we found a secondary cause for degradations. Here, most of the degradations were caused by the user saying "no" in response to the system confirming an item which is semantically *correct*, even though the user subsequently asked for the same item again. The user behavior model P_{act} – which was based on hand-crafted heuristics

TABLE IV: Causes of degradations in N-best synthesis.

Description	Instances	Percent
Degradations caused by error correlations	107	86%
Degradations not caused by error correlations	18	14%
Degradations (total)	125	100%

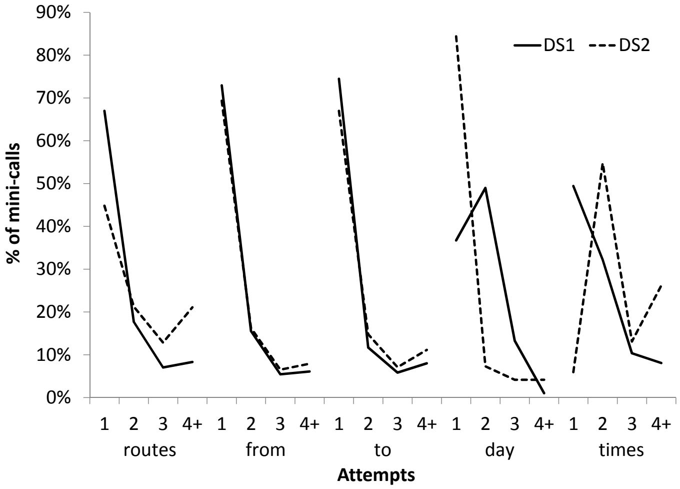


Fig. 9: Histogram of number of times each slot was requested by the system. The y axis shows the percent of each mini-call, where a mini-call is the same as a call except that "start over" begins a new mini-call.

– assigned a zero probability to this seemingly irrational behavior. As a result the correct item was ranked very low in the belief state.

Listening to these calls revealed that the confirmation wording for day was creating confusion. For example, for a call on Friday, one user said "today" but the system asked "Did you say Friday?". Similarly, another user said "tonight" and the system asked "Did you say today?" – the two are semantically identical to the system. In addition to improving this confirmation strategy, it is clear that the user action model (like the priors) can be difficult to predict and should be estimated from real usage data.

D. Confidence aggregation

Figure 5d shows that confidence aggregation had an overall positive effect, with day in DS1 being the most pronounced. The one exception was time in DS2, where there was a negative effect.

Confidence aggregation has more opportunity to occur when questions are more often asked repeatedly. Figure 9 shows histograms of how many times each slot was requested. In most cases, slots were most often requested once; however, day in DS1 and time in DS2 were usually requested more times.

Based on past investigation, we were aware that day in DS1 had a bug that set priors to be an order of magnitude too low [14]. As a result, more requests were required to obtain belief values above the (manually-set) threshold required to progress. This bug in day in DS1 was fixed in DS2. Unfortunately we found that, in the course of updating DS2, the same problem was inadvertently introduced to time in DS2. These bugs explain why these questions were more often asked repeatedly, and the disproportionately high counts of day utterances in DS1 and time utterances in DS2 in Table I.

But why was belief tracking accuracy for day in DS1 improved, whereas time in DS2 was degraded? The underlying cause was ASR re-ranking errors earlier in the dialogs. For day in DS1, ASR re-ranking yielded a small (anomalous) improvement to ASR accuracy; for time in DS2, ASR re-ranking yielded a large degradation to ASR accuracy (cf Section IV-A). Confidence aggregation amplifies these effects by carrying them forward in the dialog.

Overall this again illustrates the importance (and difficulty!) of setting model parameters correctly, and the fragility of current belief tracking methods to incorrectly specified parameters.

V. Analysis of discrimination

The analysis in the preceding sections assessed the *accuracy* of the belief state. In practice, a system must decide whether to accept or reject a hypothesis, so it is also important to evaluate the ability of the belief state to *discriminate* between correct and incorrect hypotheses. We studied this by plotting receiver operating characteristic (ROC) curves for each slot, in Figure 10. The ASR 1-best u_1 is shown using the computed $P_{\text{asr}}(u_1)$, and the top belief hypothesis s^* is shown using its belief $b(s^*)$.

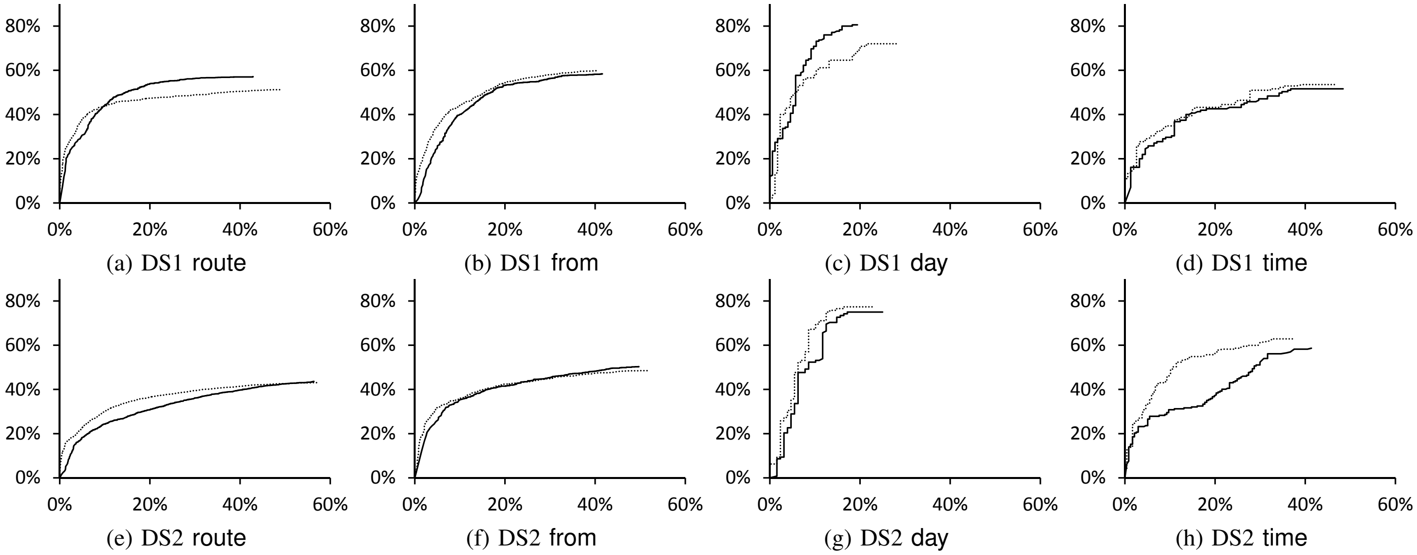
Where the belief state has markedly higher accuracy – route and day in DS1 – the belief state shows better ROC results, especially at higher false-accept rates. However, gains in ROC performance appear to be due entirely to gains in accuracy: in slots where accuracy is similar between belief tracking and ASR accuracy, the belief state shows similar or worse performance. time in DS2 was particularly affected, by the negative effect of ASR re-ranking, further compounded by confidence aggregation.

Overall, the trend appears to be that if belief tracking does not improve over ASR 1-best, then it seems that belief tracking does not enable better accept/reject decisions to be made. This suggests an important area for improvement for current techniques, and the next section suggests one avenue for future work.

VI. Conclusions and future directions

This paper has presented an in-depth analysis of 2 versions of one of the first statistical dialog systems in public use. The structure of the system – which maintains 5 independent belief states across slots with quite different properties – together with the fact that 2 versions were deployed provides a unique opportunity for analysis.

Overall, the findings have underscored the importance (and difficulty!) of correctly estimating each model component. Mismatches in all 3 component models – i.e., the models of ASR errors P_{asr}, user behavior P_{act}, and goal priors b_0 – caused degradations compared to the top speech recognition hypothesis. It is important to estimate models from data, and check the component models prior to deployment.



(a) DS1 route (b) DS1 from (c) DS1 day (d) DS1 time

(e) DS2 route (f) DS2 from (g) DS2 day (h) DS2 time

Fig. 10: ROC curves for each slot. to is very similar to from and is omitted for space. X-axis shows false accepts, Y-axis shows true accepts. The solid black line is the belief 1-best; the dotted line is the ASR 1-best. Better ROC curves tend to the upper left. The difference in each pair of curves' maximum values on the Y-axis corresponds to bar heights in Figure 3.

The systems in this paper were on-line only briefly, so there was limited opportunity to learn from interaction data. In the future, with sufficient data, it would be useful to add features to increase model accuracy. For example, the prior might benefit from adding actual bus ridership counts, the user's location, the user's past requests, the time of day, day of week, presence of special events like sports events, etc. It would also be worthwhile comparing the effects of parameters in the dialog model with parameters in the language model – the systems here used unweighted, rule-based language models. In addition, ideally all parameters (including priors) would be inferred and updated automatically from use *on-line*, and methods for doing this have been proposed [24], [25].

More broadly, the analysis here suggests fundamental weaknesses may be present in the formulation of the model, not merely in parameter estimates. For example, error correlations are not currently being modeled, and they are harming performance. The lackluster discrimination in the belief state is more troubling, suggesting that the form of the update as a generative model (Eq 1) may be problematic.

In future work, we plan to explore discriminative models for dialog state tracking [26]. Unlike the generative model analyzed in this paper (and common in the literature), discriminative methods are optimized specifically for discrimination. As such we expect they would show better ROC performance than generative models did (Figure 10). Also, discriminative methods have been suggested as being more robust to correlated ASR errors (cf [27], p 102). By including features describing past recognitions, a discriminative method may be able to learn error recurrence patterns in the data, unlike generative models which currently assume recognitions are independent. Of course, discriminative methods rely on collecting in-domain dialog data, but as this study has shown, in-domain data appears to be required to obtain good performance from generative models.

Looking ahead, the recent rise of personal assistants on mobile phones may provide an exciting new application. With an open vocabulary, multi-domain concept space, and wide-ranging noise conditions, speech recognition is challenging, creating a clear need for added robustness. Since these systems are used on a massive scale, with many repeat users, there is the potential to learn good, personalized models. This is a rich area for future work, including the challenges of scaling to a large, multi-domain (or possibly open-domain) concept space.

In sum, statistical dialog systems have seen substantial progress in the past decade, moving from toy problems to simulation and on to lab studies. The first public deployments have provided an opportunity to test whether these methods achieve their aim of improving robustness to ASR errors in practice. This paper has found that performance gains are only sometimes realized. If models are not properly estimated, there are no performance benefits. Moreover, proper estimation is deceptively difficult. On the other hand, when models *are* properly estimated – in this study, route and day in DS1 and from and to in DS2 – increased robustness to ASR errors is achieved.

APPENDIX A
DERIVATION OF UPDATE EQUATION

The belief update computes a distribution over hidden variables given the system's action, the observed ASR result, and the previous distribution over hidden variables. In this paper, two hidden variables are relevant: the user's true action u and the user's goal s. Other formulations include other components such as the dialog history [5], but these are not necessary for this paper.

Formally, we seek to estimate

$$b'(s', u') \quad = \quad P(s', u'|a, o', b).$$

where a is an observed system action, o' is an observed ASR output, s' is a hidden user goal, u' is a hidden user action,

b is the current distribution over hidden states, and b' is the updated distribution over hidden states.

Expanding using basic probability theory yields

$$\begin{aligned}
& b'(s',u') = \\
& \quad P(s',u'|a,o',b) \\
& \quad \frac{P(o'|s',u',a,b)p(s',u'|a,b)}{P(o'|a,b)} \\
& \quad \frac{P(o'|s',u',a,b)\sum_s \sum_u P(s',u'|s,u,a,b)P(s,u|a,b)}{P(o'|a,b)} \\
& \quad \frac{P(o'|s',u',a,b)\sum_s \sum_u P(s',u'|s,u,a)b(s,u)}{P(o'|a,b)}.
\end{aligned}$$

o', a, and b are fixed for any s', and can be written as a constant k

$$k \quad = \quad \frac{1}{P(o'|a,b)}.$$

Substituting,

$$\begin{aligned}
& b'(s',u') = \\
& \quad k \cdot P(o'|s',u',a,b) \sum_s \sum_u P(s',u'|s,u,a)b(s,u).
\end{aligned}$$

The ASR result depends only on the user's action $P(o'|s',u',a,b) = P(o'|u')$:

$$b'(s',u') \quad = \quad k \cdot P(o'|u') \sum_s \sum_u P(s',u'|s,u,a)b(s,u).$$

Decomposing $P(s',u'|s,u,a)$ yields

$$\begin{aligned}
& b'(s',u') = \\
& \quad k \cdot P(o'|u') \sum_s \sum_u P(s'|s,u,a)P(u'|s',s,u,a)b(s,u).
\end{aligned}$$

We assume that the user's new goal s' depends only on their previous goal s and the system action a, and that the user's action u' depends only on their (new) goal s' and the system's action a:

$$\begin{aligned}
b'(s',u') &= k \cdot P(o'|u') \sum_s \sum_u P(s'|s,a)P(u'|s',a)b(s,u) \\
&= k \cdot P(o'|u') \sum_s P(s'|s,a)P(u'|s',a) \sum_u b(s,u) \\
&= k \cdot P(o'|u')P(u'|s',a) \sum_s P(s'|s,a)b(s)
\end{aligned}$$

In the dialog systems in this paper, actions are chosen based on the (marginal) distribution over user goals s'; marginalizing the user's action u' yields

$$\begin{aligned}
\sum_{u'} b'(s',u') &= \sum_{u'} k \cdot P(o'|u')P(u'|s',a) \sum_s P(s'|s,a)b(s) \\
b'(s') &= k \cdot \sum_{u'} P(o'|u')P(u'|s',a) \sum_s P(s'|s,a)b(s)
\end{aligned}$$

Also, in this dialog system, it is assumed that the user's goal s' is fixed – i.e., $P(s'|s,a) = \delta(s',s)$, where δ is the Kronecker delta function:

$$\begin{aligned}
b'(s') &= k \cdot \sum_{u'} P(o'|u')P(u'|s',a) \sum_s \delta(s',s)b(s) \\
b'(s') &= k \cdot \sum_{u'} P(o'|u')P(u'|s',a)b(s')
\end{aligned}$$

Finally, note that $P(o'|u')$ is a generative model of ASR results, which is difficult to estimate. So in practice $P(o'|u')$ is usually re-written

$$P(o'|u') \quad = \quad \frac{P(u'|o')P(u')}{P(o')}.$$

It is then assumed that $P(u')$ is uniform. While this is not strictly true, it is reasonable given that the update already includes $P(u'|s',a)$, which is more specific than $P(u')$. Thus $P(u')$ and $P(o')$ are both constant, so we can write:

$$P(o'|u') \quad \approx \quad \eta P(u'|o'),$$

where η is a constant. Substituting into the update:

$$b'(s') \quad = \quad k \cdot \sum_{u'} P(u'|o')P(u'|s',a)b(s')$$

where k is still a normalization constant, although its definition has changed. Dropping the (now unnecessary) prime from variables and labeling the two probability models yields the update in Eq 1

$$b'(s) \quad = \quad k \cdot \sum_u P_{\text{asr}}(u)P_{\text{act}}(u|s,a)b(s).$$

APPENDIX B
MECHANISM DETAIL

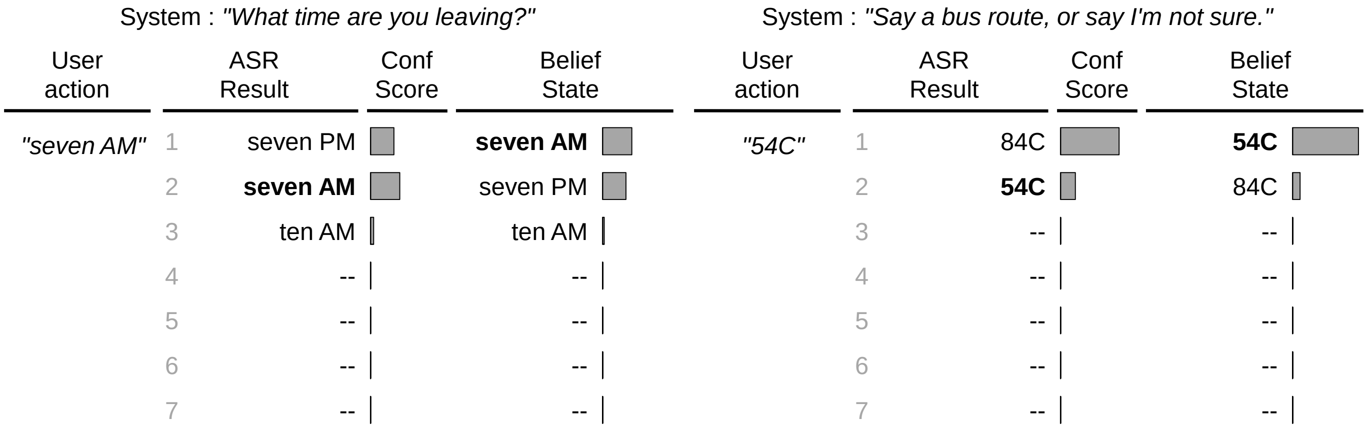
Figure 11 provides graphical illustrations of each of the four *mechanisms* that can cause the top ASR hypothesis u_1 to be different from the top belief state hypothesis s^*. These examples were taken from logs of calls with real users, although some surface forms have been simplified for space.

At the top of each panel is the system action taken. The user's true response is shown in italics in the left-most column. The second column shows the top 7 entries from the ASR N-best list, displayed in the order produced by the speech recognition engine. The third column shows the confidence score – the local probability of correctness assigned to each ASR N-best entry by P_{asr}. The last column shows the resulting belief state $b(s^*)$, sorted by the magnitude of the belief. Correct entries are shown in bold.

ASR re-ranking and prior re-ranking occur within one turn, and confidence aggregation and N-best synthesis occur across two turns. These examples all show cases where the belief state is correct and the ASR is incorrect; however, the opposite also occurs of course.

ACKNOWLEDGMENT

The author thanks Alan Black and the Dialog Research Center at CMU for hosting the Spoken Dialog Challenge, and for making the log data available. The author also thanks collaborator Angeliki Metallinou for sharing early results of discriminative approaches to belief tracking, Alex Acero for insightful discussions, and Frank Seide for helpful comments on presentation.



(a) **Illustration of ASR re-ranking**: The correct ASR hypothesis ("seven AM") is in the $n = 2$ position, but it is assigned a higher confidence score than the misrecognized $n = 1$ entry "seven PM". time uses a flat prior, so the higher confidence score results in "seven AM" attaining the highest belief.

(b) **Illustration of Prior re-ranking**: The correct ASR hypothesis ("54C") is in the $n = 2$ position, and it is assigned less confidence by P_{asr} than the mis-recognized $n = 1$ entry, "84C". However, the prior b_0 on 54C is much higher than on 84C, so 54C obtains the highest belief.

Fig. 11: Examples of the ASR re-ranking and prior re-ranking mechanisms. See below for Figures 11c and 11d.

References

[1] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialogue strategies," *IEEE Trans on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[2] T. Paek and E. Horvitz, "Conversation as action under uncertainty," in *Proc Conf on Uncertainty in Artificial Intelligence (UAI), Stanford, California*, 2000, pp. 455–464.

[3] B. Zhang, Q. Cai, J. Mao, and B. Guo, "Planning and acting under uncertainty: A new model for spoken dialogue system," in *Proc Conf on Uncertainty in Artificial Intelligence (UAI), Seattle, Washington*, 2001, pp. 572–579.

[4] N. Roy, J. Pineau, and S. Thrun, "Spoken dialog management for robots," in *Proc Association for Computational Linguistics (ACL), Hong Kong*, 2000, pp. 93–100.

[5] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.

[6] T. Bui, M. Poel, A. Nijholt, and J. Zwiers, "A tractable hybrid DDN-POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems," *Natural Language Engineering*, vol. 15, no. 2, pp. 273–307, 2009.

[7] J. D. Williams, "Incremental Partition Recombiantion for Efficient Tracking of Multiple Dialogue States." in *ICASSP*, Dallas, TX, 2010.

[8] H. Higashinaka, M. Nakano, and K. Aikawa, "Corpus-based discourse understanding in spoken dialogue systems," in *Proc Association for Computational Linguistics (ACL), Sapporo, Japan*, 2003.

[9] J. Henderson and O. Lemon, "Mixture model POMDPs for efficient handling of uncertainty in dialogue management," in *Proc Association for Computational Linguistics Human Language Technologies (ACL-HLT), Columbus, Ohio*, 2008.

[10] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2009.

[11] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[12] A. W. Black, S. Burger, A. Conkie, H. Hastie, S. Keizer, O. Lemon, N. Merigaud, G. Parent, G. Schubiner, B. Thomson, J. D. Williams, K. Yu, S. Young, and M. Eskenazi, "Spoken dialog challenge 2010: Comparison of live and control test results," in *Proc SIGdial Workshop on Discourse and Dialogue, Portland, Oregon*, 2011.

[13] J. D. Williams, I. Arizmendi, and A. Conkie, "Demonstration of AT&T "Let's Go": A production-grade statistical spoken dialog system," in *Proc Workshop on Spoken Language Technologies (SLT), Berkeley, California, USA*, 2010.

[14] J. D. Williams, "An Empirical Evaluation of a Statistical Dialog System in Public Use," in *Proc SIGdial Workshop on Discourse and Dialogue, Portland, Oregon*, 2011.

[15] M. Gasic and S. Young, "Effective handling of dialogue state in the hidden information state POMDP-based dialogue manager," *ACM Transactions on Speech and Language Processing*, 2011.

[16] K. Kim, C. Lee, S. Jung, and G. Lee, "A Frame-Based Probabilistic Framework for Spoken Dialog Management Using Dialog Examples," Columbus, Ohio, 2008, pp. 120–127.

[17] N. Mehta, R. Gupta, A. Raux, D. Ramachandran, and S. Krawczyk, "Probabilistic ontology trees for belief tracking in dialog systems," in *Proc SIGdial Workshop on Discourse and Dialogue, Tokyo, Japan*, 2010.

[18] J. D. Williams, "Using particle filters to track dialogue state," in *Proc IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Kyoto, Japan*, 2007.

[19] A. W. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi, "Spoken dialog challenge 2010," in *Proc Workshop on Spoken Language Technologies (SLT), Berkeley, California, USA*, 2010.

[20] "Dialog Research Center, Carnegie Mellon University," dialrc.org.

[21] "AT&T Statistical Dialog Toolkit," http://www2.research.att.com/sw/tools/asdt/.

[22] J. D. Williams and S. Balakrishnan, "Estimating probability of correctness for ASR N-Best lists," in *Proc SIGdial Workshop on Discourse and Dialogue, London, UK*, 2009.

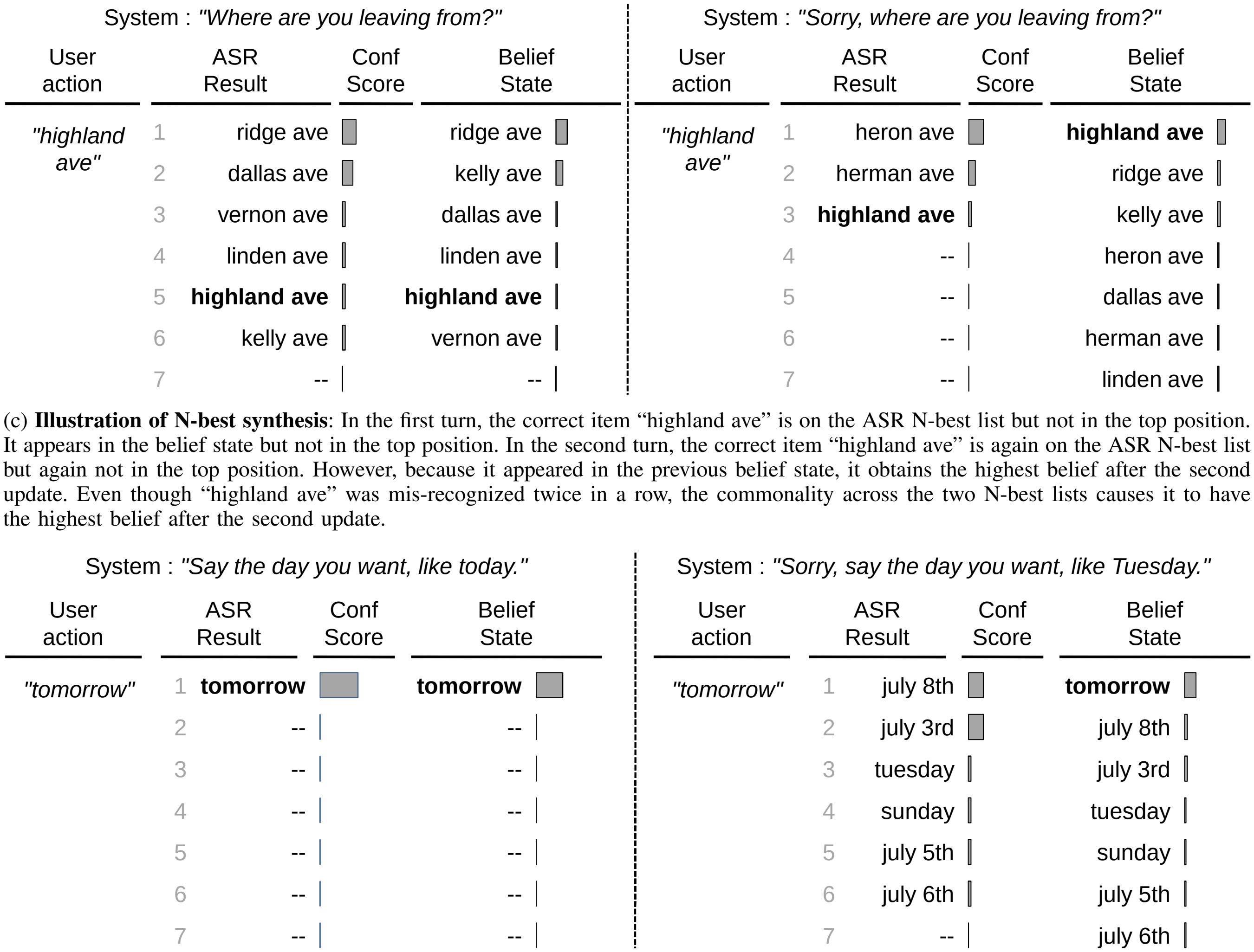
[23] G. Parent and M. Eskenazi, "Toward Better Crowdsourced Transcription: Transcription of a Year of the Let's Go Bus Information System Data," in *Proc Workshop on Spoken Language Technologies (SLT), Berkeley, California, USA*, 2010.

[24] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann, "A Bayesian approach for learning and planning in partially observable Markov decision processes," *Journal of Machine Learning*, vol. 12, pp. 1655–1696, 2011.

[25] F. Jurcicek, B. Thomson, and S. Young, "Reinforcement learning for parameter estimation in statistical spoken dialogue systems," *Computer Speech and Language*, 2012.

[26] D. Bohus and A. Rudnicky, "A 'K hypotheses + other' belief updating model," in *Proc American Association for Artificial Intelligence (AAAI) Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems, Boston*, 2006.

[27] D. Bohus, "Error awareness and recovery in conversational spoken language interfaces," Ph.D. dissertation, Carnegie Mellon University, 2007.



System : *"Where are you leaving from?"*

User action		ASR Result	Conf Score	Belief State	
"highland ave"	1	ridge ave	▮	ridge ave	▮
	2	dallas ave	▮	kelly ave	▮
	3	vernon ave	▮	dallas ave	▮
	4	linden ave	▮	linden ave	▮
	5	**highland ave**	▮	**highland ave**	▮
	6	kelly ave	▮	vernon ave	▮
	7	--	▮	--	▮

System : *"Sorry, where are you leaving from?"*

User action		ASR Result	Conf Score	Belief State	
"highland ave"	1	heron ave	▮	**highland ave**	▮
	2	herman ave	▮	ridge ave	▮
	3	**highland ave**	▮	kelly ave	▮
	4	--	▮	heron ave	▮
	5	--	▮	dallas ave	▮
	6	--	▮	herman ave	▮
	7	--	▮	linden ave	▮

(c) **Illustration of N-best synthesis**: In the first turn, the correct item “highland ave” is on the ASR N-best list but not in the top position. It appears in the belief state but not in the top position. In the second turn, the correct item “highland ave” is again on the ASR N-best list but again not in the top position. However, because it appeared in the previous belief state, it obtains the highest belief after the second update. Even though “highland ave” was mis-recognized twice in a row, the commonality across the two N-best lists causes it to have the highest belief after the second update.

System : *"Say the day you want, like today."*

User action		ASR Result	Conf Score	Belief State	
"tomorrow"	1	**tomorrow**	▮	**tomorrow**	▮
	2	--	▮	--	▮
	3	--	▮	--	▮
	4	--	▮	--	▮
	5	--	▮	--	▮
	6	--	▮	--	▮
	7	--	▮	--	▮

System : *"Sorry, say the day you want, like Tuesday."*

User action		ASR Result	Conf Score	Belief State	
"tomorrow"	1	july 8th	▮	**tomorrow**	▮
	2	july 3rd	▮	july 8th	▮
	3	tuesday	▮	july 3rd	▮
	4	sunday	▮	tuesday	▮
	5	july 5th	▮	sunday	▮
	6	july 6th	▮	july 5th	▮
	7	--	▮	july 6th	▮

(d) **Illustration of Confidence aggregation**: In the first turn, “tomorrow” is recognized with medium confidence. In the second turn, “tomorrow” does not appear on the N-best list; however the recognition result has very low confidence, so this misrecognition is unable to dislodge “tomorrow” from the top belief position. At the end of the second update, the belief state’s top hypothesis of “tomorrow” is correct even though it didn’t appear on the second N-best list.

Fig. 11: Examples of the N-best synthesis and confidence aggregation mechanisms. See above for Figures 11a and 11b.

Jason Williams Jason Williams is with Microsoft Research. His interests include spoken dialog systems, planning under uncertainty, spoken language understanding, and speech recognition, and he has published 40 papers in these areas. He is on the Scientific Committee of SigDial (the Special Interest Group on Dialog and Discourse) and is on the board of directors of the Association for Voice Interaction Design (AVIxD). From 2009-2011, he served on the IEEE Speech and Language Technical Committee (SLTC) in the area of spoken dialogue systems. He holds a PhD and Masters in Speech and Language Processing from Cambridge University (UK), and a BSE in Electrical Engineering from Princeton University (USA). Prior to Microsoft, Jason was Principal Member of Technical Staff at AT&T Labs – Research from 2006-2012. Jason has also held several positions in industry building spoken dialog systems, including at Tellme Networks (now Microsoft) as Voice Application Development Manager. Systems he has deployed over his career have engaged in over 100 million dialogs with real users.