

The Dialog State Tracking Challenge Series: A Review

Jason D. Williams

Microsoft Research
One Microsoft Way
Redmond, WA, 98052, USA

JASON.WILLIAMS@MICROSOFT.COM

Antoine Raux

Facebook
1 Facebook Way
Menlo Park, CA 94025, USA

ARAUX@FB.COM

Matthew Henderson*

Department of Engineering
University of Cambridge
CB2 1PZ, UK

MATTHEN@GOOGLE.COM

Editor: David Schlangen

Submitted 04/15; Accepted 02/16; Published online 04/16

Abstract

In a spoken dialog system, *dialog state tracking* refers to the task of correctly inferring the state of the conversation – such as the user’s goal – given all of the dialog history up to that turn. Dialog state tracking is crucial to the success of a dialog system, yet until recently there were no common resources, hampering progress. The *Dialog State Tracking Challenge* series of 3 tasks introduced the first shared testbed and evaluation metrics for dialog state tracking, and has underpinned three key advances in dialog state tracking: the move from generative to discriminative models; the adoption of discriminative sequential techniques; and the incorporation of the speech recognition results directly into the dialog state tracker. This paper reviews this research area, covering both the challenge tasks themselves and summarizing the work they have enabled.

Keywords: Dialog state tracking, spoken dialog systems, dialog modeling, conversational systems, spoken language understanding

1. Introduction

Conversational systems are increasingly becoming a part of daily life, with examples including Apple’s Siri, Google Now, Nuance Dragon Go, Xbox and Cortana from Microsoft, and numerous start-ups. Figure 1 shows the principal components of a modern spoken dialog system. First, the user produces an utterance as audio. Then *automatic speech recognition* (ASR) converts this audio into words in text form. Next, the words in an utterance are converted to a meaning representation using *spoken language understanding* (SLU). This SLU result is then passed to the *dialog state tracker* (DST) which updates its estimate of the dialog state. This new dialog state is passed to the dialog policy that decides which action to take. *Natural language generation* (NLG) and *text-to-speech* (TTS) convert this action into words and then into audio. The cycle then repeats.

. * Matthew Henderson is now at Google.

DIALOG STATE TRACKING OVERVIEW

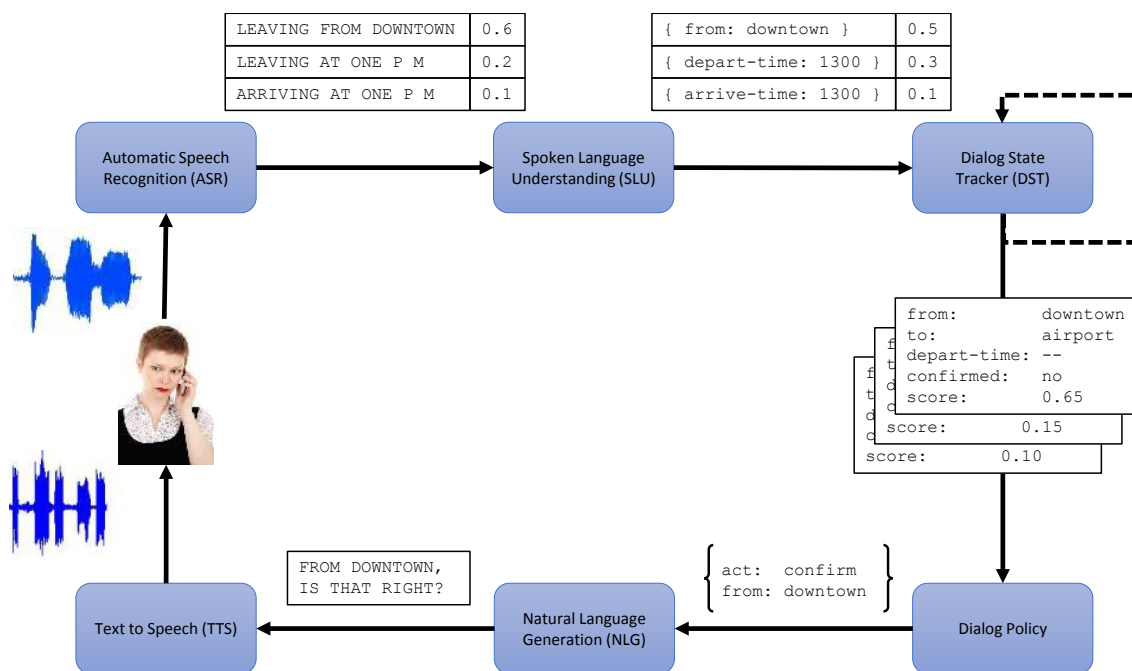


Figure 1: Principal components of a spoken dialog system.

The topic of this paper is the *dialog state tracker* (DST). The DST takes as input all of the dialog history so far, and outputs its estimate of the current *dialog state* – for example, in a restaurant information system, the dialog state might indicate the user’s preferred price range and cuisine, what information they are seeking such as the phone number of a restaurant, and which concepts have been stated vs. confirmed. Dialog state tracking is difficult because ASR and SLU errors are common, and can cause the system to misunderstand the user. At the same time, state tracking is crucial because the dialog policy relies on the estimated dialog state to choose actions – for example, which restaurants to suggest.

In the literature, numerous methods for dialog state tracking have been proposed. These are covered in detail in Section 3; illustrative examples include hand-crafted rules (Larsson and Traum, 2000; Bohus and Rudnicky, 2003), heuristic scores (Higashinaka et al., 2003), Bayesian networks (Paek and Horvitz, 2000; Williams and Young, 2007), and discriminative models (Bohus and Rudnicky, 2006). Techniques have been fielded which scale to realistically sized dialog problems and operate in real time (Young et al., 2010; Thomson and Young, 2010; Williams, 2010; Mehta et al., 2010). In end-to-end dialog systems, dialog state tracking has been shown to improve overall system performance (Young et al., 2010; Thomson and Young, 2010).

Despite this progress, direct comparisons between methods have not been possible because past studies use different domains and different system components for ASR, SLU, dialog policy, etc. Moreover, there has not been a standard task or methodology for evaluating dialog state tracking. Together these issues have limited progress in this research area.

The Dialog State Tracking Challenge (DSTC) series has provided a first common testbed and evaluation suite for dialog state tracking. Three instances of the DSTC have been run over a three

year period. Each instance has released a public corpus of transcribed and labeled human-computer dialogs along with baseline trackers and evaluation tools, and each instance has explored a new aspect of dialog state tracking. Between seven and nine teams have entered each challenge. This challenge task series has spurred significant work on dialog state tracking, yielding both numerous new techniques as well as a standard set of evaluation metrics.

This paper is organized as follows. First, Section 2 formalizes the dialog state tracking problem, and Section 3 reviews solution methods from the literature. Section 4 then covers the first three instances of the dialog state tracking challenge – **DSTC1**, **DSTC2**, and **DSTC3** – including the task design, data, evaluation methodology, and baselines. Section 5 then covers results from the challenge tasks. Finally, section 7 concludes.

2. Dialog state tracking: problem definition

First, we define the concept of *dialog state*. A dialog state s_t is a data structure drawn from a set S that summarizes the dialog history up to time t to a level of detail that provides sufficient information for choosing the next system action. In practice, the dialog state typically encodes the user’s goal in the conversation along with relevant history – for example, in the bus timetable domain, s may encode which bus stop the user wants to leave from, where they are going to, and whether the system has already offered a bus on that route.

A *dialog state tracker* takes as input all of the observable elements up to time t in a dialog, including all of the results from the ASR and SLU components, all system actions taken so far, and external knowledge sources such as bus timetable databases and models of past dialogs. Because the ASR and SLU are imperfect and prone to errors, they may output several conflicting interpretations. Specifically, the ASR may output an N-Best list of sentences, a word confusion network (Mangu et al., 2000), or a lattice; the SLU may output an N-Best list of interpretations. Figure 1 shows example ASR and SLU N-Best lists.

Given these inputs, the tracker then outputs its estimate of the current state of the dialog s . The goal is to correctly identify the true current state s^* of the dialog – for example, the bus stops the user has actually said they want or whether the user wants the address, opening hours, or price range of a particular restaurant. However, the true state is typically not directly observable from the inputs, for a variety of reasons: errors in speech recognition and language understanding, ambiguous or underspecified utterances, unsignaled changes in the user’s goal, etc. Therefore, robust dialog state trackers typically output a *distribution* over multiple possible dialog states $b(s)$. A distribution is useful because it provides a principled representation of the uncertainty in the dialog state. It also gives a clear basis for taking clarification actions: for example, if the distribution’s probability mass is concentrated on two states that differ only in which type of food the user is asking for (say, “Indian” and “Italian”), this allows the system to ask “Did you want Indian or Italian food?”. Figure 2 shows an example of the dialog state tracking process, and illustrates how effective dialog state tracking can overcome some of the errors received from the ASR and SLU.

In this paper – and in the DSTC challenge series – we have taken the view that a dialog state consists of elements with human-interpretable meanings, such as values of bus stops, dates, times, whether conditions have been met, etc. We have further assumed that a dialog state tracker produces the key input to an action selector – sometimes also called a “dialog policy” – that chooses an action or response based primarily on the current dialog state. This view is in line with widely accepted theoretical models of conversation, such as Clark’s Common Ground (Clark, 1996) and various

DIALOG STATE TRACKING OVERVIEW

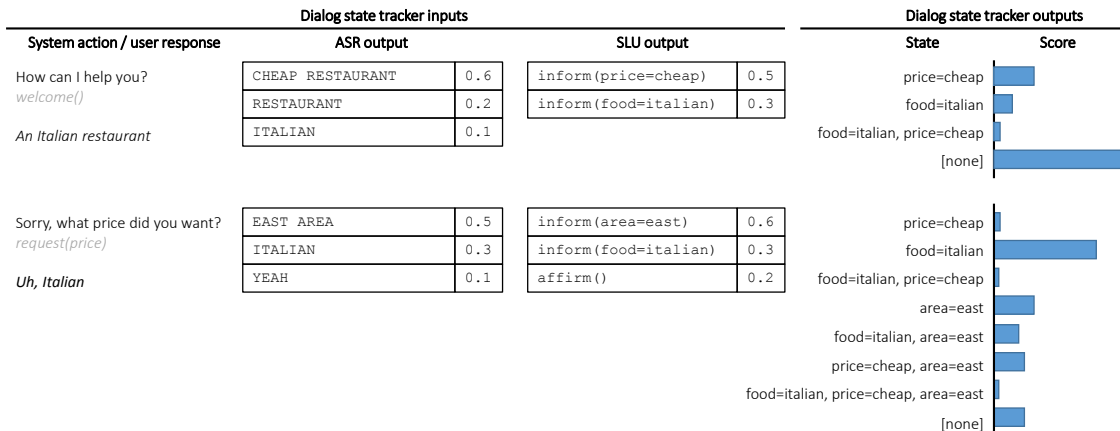


Figure 2: Overview of dialog state tracking. In this example, the dialog state contains the user’s desired restaurant search criteria. At each turn, the system produces a spoken output. The user’s spoken response is converted into an N-best list of word hypotheses by the ASR, and then into another N-Best list of meaning hypotheses by the SLU. Both lists have confidence scores attached. A set of dialog state hypotheses is enumerated, here by simply considering all SLU results observed so far, including the current turn and all previous turns. Then the dialog states are scored. Note how observing “italian” a second time in the ASR/SLU causes the dialog state for “food=italian” to accumulate considerable probability mass in the second turn, even through “italian” was never the top hypothesis from the ASR or SLU. This illustrates one way that dialog state tracking can overcome local ASR/SLU errors.

models of dialog as joint action (Cohen and Levesque, 1990), which assume that dialog relies on some (usually shared) representation of the participants’ joint intentions and beliefs. While this is the dominant approach, it is worth mentioning alternatives. First, dialog state can instead be a latent representation, with responses selected – or in principle generated – using continuous-space representations (Lowe et al., 2015). Further, it is possible to dispense with state tracking altogether, and instead produce responses based only on the most recent user turn (Ritter et al., 2011) – or in principle directly from features of the dialog history. A comparison with these methods would require end-to-end evaluations of spoken dialog systems, which is outside the scope of the DSTC series, and this paper.¹

In the next section we review methods for dialog state tracking.

3. Methods for dialog state tracking

Broadly speaking there are three families of dialog state tracking algorithms: *hand-crafted rules*, *generative models*, and *discriminative models*.

1. Dialog state tracking in *situated* environments – for example, robots or embodied agents – is also out of scope for this review, but it is noted that dialog state tracking is also used in this setting (Bohus and Horvitz, 2009; Ma et al., 2012).

3.1 Hand-crafted rules for dialog state tracking

Early spoken dialog systems used *hand-crafted rules* for dialog state tracking. In their earliest form, these approaches generally considered only a single SLU result, and tracked a single hypothesis for the dialog state. This design reduces the dialog state tracking problem to an update rule $F(s, \tilde{u}') = s'$ that maps from an existing state s and the 1-best SLU result \tilde{u}' to a new state s' . For example, the MIT JUPITER weather information system maintained a set of state variables which were updated using hand-written rules in a dialog control table (Zue et al., 2000). Similarly, the Information State Update approach used hand-written update rules to track a rich data structure called an “information state” (Larsson and Traum, 2000).

Hand-crafted rules have the benefit that they do not require any data to implement, which is a benefit for bootstrapping. Rules also provide an accessible way for developers to incorporate domain knowledge. However, one short-coming of tracking a single dialog state is an inability to make use of the entire ASR or SLU N-Best list, and the benefit of tracking multiple dialog states was suggested nearly two decades ago by Pulman (1996). Thus, more recent dialog state trackers based on hand-crafted rules compute scores for all dialog states suggested by the whole ASR/SLU N-best list (Wang and Lemon, 2013; Sun et al., 2014a). These methods use hand-designed formulas to compute a posterior $b(s)$ of a dialog state s given ASR/SLU confidence scores and previous estimates of $b(s)$, and thus can overcome some SLU errors (Figure 2).

Using hand-designed formulas for computing $b(s)$ suffers from a crucial limitation: formula parameters are not derived directly from real dialog data, so they require careful tuning and do not benefit or learn from dialog data. This limitation motivates the use of data-driven techniques, which can automatically set parameters in order to maximize accuracy. Chief among the data-driven techniques are *generative* and *discriminative* models, described next.

3.2 Generative models for dialog state tracking

Generative approaches posit that dialog can be modeled as a Bayesian network that relates the dialog state s to the system action a , the (true, unobserved) user action u , and ASR or SLU result \tilde{u} . When the system action and ASR/SLU result are observed, a distribution over possible dialog states can be computed by applying Bayesian inference. A number of probabilistic formulations have been explored for how to relate these quantities; one illustrative example is:

$$b'(s') = \eta \sum_{u'} P(\tilde{u}'|u') P(u'|s', a) \sum_s P(s'|s, a) b(s) \quad (1)$$

where $b(s)$ is the previous distribution over dialog states, $b'(s')$ is the (updated) distribution over dialog states being estimated, $P(\tilde{u}'|u')$ is the probability of the ASR/SLU producing the observed output \tilde{u}' given the (true, unobserved) user action u' , $P(u'|s', a)$ is the probability of the user taking action u' given the true dialog state s' and system action a , $P(s'|s, a)$ is probability of the dialog state changing to s' given it is currently s and the system takes action a , and η is a normalizing constant.

Variants of Eq. 1 account for different factorizations of the hidden state. For example, Williams and Young (2007) includes a term that accumulates dialog history, such as whether the contents of s has been confirmed or not. DeVault and Stone develop a Bayesian network that includes separate random variables for an observed dialog action and an underlying intention, and includes conditional probability terms that express common-sense relationships between actions, intentions, and

plausible states termed “contexts” (DeVault and Stone, 2007; DeVault, 2008). Other factorizations have also been presented for modeling dialog in specific settings, such as troubleshooting an internet router (Williams, 2007). Eq. 1 most closely follows Williams (2008); the appendix of Williams (2012a) provides a derivation. In all of these examples, the key assumption is that a distribution over possible (hidden) dialog states can be inferred using a Bayesian network that encodes a designer’s knowledge about conversation. The parameters of the models must be estimated of course; this can be done either from labeled dialogs, or inferred from unlabeled dialogs using methods such as Expectation Maximization (Syed and Williams, 2008) or Expectation Propagation (Thomson et al., 2010).

Early approaches to generative dialog state tracking enumerated all possible dialog states, then used variants of Eq. 1 to score them (Roy et al., 2000; Zhang et al., 2001; Heckerman and Horvitz, 1998; Horvitz and Paek, 1999; Meng et al., 2003; Williams et al., 2005). This approach is quadratic in the number of dialog states, which is intractable, particularly given that Eq. 1 must run in real time and the number of states s can be enormous. This limitation has led to two approximations: maintaining a “beam” of only the most likely members of s (Young et al., 2007; DeVault and Stone, 2007; DeVault, 2008; Kim et al., 2008; Henderson and Lemon, 2008; Mehta et al., 2010; Williams, 2010; Raux and Ma, 2011; Gasic and Young, 2011), or further factorization of Eq. 1 (Williams, 2007; Bui et al., 2009; Thomson and Young, 2010). These approximations enable generative models to operate in real-time, but impose other constraints, such as limiting the form of $P(s'|s, a)$, which can restrict the classes of dialogs that can be accurately modeled (Young et al., 2013).

In end-to-end evaluations, generative approaches have been shown to yield better dialog performance than hand-crafted rules (Young et al., 2010; Thomson and Young, 2010). Even so, generative models cannot easily incorporate large sets of potentially informative features from the ASR, SLU, dialog history, and elsewhere: all dependencies between features must be explicitly modeled, which requires an impractical amount of data. As a result, for tractability, generative models generally make independence assumptions which are invalid, or important features of dialog history have to be ignored, which introduce violation of the Markov assumption. For example, it is often assumed that errors are generated from a uniform distribution, when in fact they are highly correlated: “twenty” is much more often mis-recognized as “seventy” than as “downtown pittsburgh” (Williams, 2012c). The net effect is poor estimates of $b(s)$. Together these issues have spurred interest in discriminatively trained direct models, covered next.

3.3 Discriminative models for dialog state tracking

In contrast to generative models, discriminative approaches to dialog state tracking compute scores for dialog states with discriminatively trained conditional models of the form $b'(s') = P(s'|\mathbf{f}')$, where \mathbf{f}' are features extracted from the ASR, SLU, and dialog history. The key benefit of discriminative models are that they can incorporate a large number of features, and can be optimized directly for prediction accuracy.

The first presentation of discriminative state tracking trained from data is believed to be Bohus and Rudnicky (2006). Here, a hand-written rule enumerates a set of k dialog states to score, for example by considering the top S_1 SLU hypotheses from the current turn, top S_2 SLU hypotheses from the previous turn, and the top S_3 SLU hypothesis from the turn before that. An additional state hypothesis \bar{s} accounts for the situation when none of the hypotheses is correct, for a total of $k = S_1 + S_2 + S_3 + 1$ states to score. With a fixed number k of classes, standard multiclass logistic

regression classification is then applied, in which one weight is estimated for every (class,feature) pair. Features were taken from SLU output and dialog history.

Subsequent work has explored numerous variations of this approach. Metallinou et al. (2013) alter the logistic regression model so that it learns a single weight for each feature. This enables an arbitrary number of hypotheses to be scored since the number of weights to learn no longer increases with the number of state hypotheses to score. Williams (2014) applies a ranking algorithm which has the ability to construct conjunctions of features. Henderson et al. (2013) applies a deep neural network as a classifier.

All of the approaches above encode dialog history in the features to learn a simple classifier. By contrast, three other approaches have explicitly modeled dialog as a *sequential* process. First, a discriminative Markov Model can be applied, where the distribution from the previous turn’s prediction can be used as a feature (Ren et al., 2014b,a). Second, dialog can be cast as a *conditional random field* (CRF) (Lafferty et al., 2001), in which features are associated with each dialog turn, and CRF decoding determines the most likely final dialog state conditioned on the entire sequence (Lee and Eskenazi, 2013; Ren et al., 2013; Kim and Banchs, 2014; Ma and Fosler-Lussier, 2014c). Third, recurrent neural networks can be estimated where the inputs are the observed ASR/SLU results, and the output is a distribution over dialog states (Henderson et al., 2014d).

Henderson et al. (2014d) is also notable for operating directly on ASR output, without an SLU (c.f. Figure 1). This has two benefits: first, it removes the need for feature design, and the risk of omitting an important feature, which can degrade performance unexpectedly (Williams, 2014). Second, it avoids the work of building a separate SLU model.

All of the approaches above require in-domain dialog data for training. When a small amount of labeled data exists for the target domain, multi-domain learning can be applied (Williams, 2013). When no labeled data exists – for example, when a system is first deployed – it is possible to use unsupervised adaptation from a base model for a related domain. The basic idea is to find points in the dialog where a state component value is assigned a high score – such as `food=italian` – then treat that predicted value as a label, and adjust model parameters to predict that label earlier in the dialog (Lee and Eskenazi, 2013; Henderson et al., 2014e). This approach allows a generic slot tracking model to be adapted to a specific slot for which labeled data does not exist.

The approaches above infer user behavior directly from the dialog data, and make no a priori assumptions about the structure of $P(s'|f')$. Since some properties of human behavior with dialog systems is known – for example, that people typically change their goal only in certain situations – it is possible to devise rules that score dialog states using functions of the ASR or SLU confidence scores, and then estimate a handful of parameters of the rules from data (Higashinaka et al., 2003; Kadlec et al., 2014; Sun et al., 2014a). Since the primary source of uncertainty in dialog state tracking is the ASR or SLU, these methods can perform very well when the confidence scores are reliable.

With so many methods for dialog state tracking proposed, it is vital to have benchmark tasks for making performance comparisons. This need motivated the *Dialog State Tracking Challenge* series of research community tasks, described next.

4. Challenge tasks

4.1 Overview

The over-arching research aim of the DSTC series has been to understand which existing methods for dialog state tracking perform best, and encourage new work that advances the state-of-the-art. As part of that aim, the **DSTC** series has also examined which evaluation measurements are appropriate for dialog state tracking.

To date there have been three completed dialog state tracking challenges. Each has used logs of human-computer dialogs in different domains, with different properties:

DSTC1 used a corpus of dialogs with various systems that participated in the Spoken Dialog Challenge (SDC) (Black et al., 2010), provided by the Dialog Research Center at Carnegie Mellon University. In the SDC, telephone calls from real passengers of the Port Authority of Allegheny County, which runs city buses in Pittsburgh, were forwarded to dialog systems built by different research groups. The goal was to provide bus riders with bus timetable information. For example, a caller might want to find out the time of the next bus leaving from Downtown to the airport. In this domain, the goal of the user typically remains fixed for the duration of the dialog.

DSTC2 aimed to extend the results of **DSTC1** to another domain, as well as broaden the scope to include user goal changes. This challenge relied on a corpus of dialogs in the restaurant search domain between paid participants (through Amazon Mechanical Turk) and various systems developed at Cambridge University (Young et al., 2014). The goal of the user is to find specific information such as price range or phone number about a restaurant that fulfills a number of constraints such as cuisine or neighborhood.

DSTC3 expanded the domain of **DSTC2** to include new slots which do not occur in the training data. This simulates the crucial problem of adapting a dialog system to a new domain for which little dialog data is available, while data for a similar but different domain might already exist. **DSTC3** used all the data from **DSTC2** as training set, as well as a new set of dialogs (also collected by Cambridge University researchers (Jurčiček et al., 2011)) on a broader tourist information domain, covering bars and cafes in addition to restaurants.

4.2 Challenge Design

The dialog state tracking challenges take a *corpus-based* approach – i.e., dialog state trackers are trained and tested on a *static corpus of dialogs*, recorded from systems using a variety of state tracking models and dialog managers. The challenge task is to *re-run* state tracking on these dialogs – i.e., to take as input the runtime system logs including the SLU results and system output, and to output scores for dialog states. This corpus-based design was chosen because it allows different trackers to be evaluated on the same data, and because a corpus-based task has a much lower barrier to entry for research groups than building an end-to-end dialog system.

In practice of course, a state tracker will be used in an end-to-end dialog system, and will drive action selection, thereby affecting the distribution of the dialog data the tracker experiences. In other words, it is known in advance that the distribution in the training data and live data will be mismatched, although the nature and extent of the mis-match are not known. Hence, unlike much of supervised learning research, drawing train and test data from the same distribution in offline experiments may overstate performance. So in all three challenges, train/test mis-match was

explicitly created by choosing test data to be from different dialog systems, and, in the case of **DSTC3**, with a different set of slots to be filled.

4.3 Data

The corpus for **DSTC1** was produced with dialog systems from three different research groups, here called Groups A, B, and C. Each group used its own ASR, SLU, and dialog manager. The dialog strategies across groups varied considerably: for example, Groups A and C used a mixed-initiative design, where the system could recognize any concept at any turn, but Group B used a directed design, where the system asked for concepts sequentially and could only recognize the concept being queried. Groups trialed different system variants over a period of almost 3 years. These variants differed in acoustic and language models, confidence scoring model, state tracking method and parameters, number of supported bus routes, user population, and presence of minor bugs. The fact that these systems were actually deployed and used by the general public presented a number of challenges, most notably acoustic and linguistic conditions made ASR significantly more difficult than in more controlled settings. The average length of a dialog in **DSTC1** is 14.1 turns. More descriptive statistics are given in Table 1. **DSTC1** released 5 train sets and 4 test sets. In all train sets, user speech was transcribed, but only 3 of the 5 train sets were labeled for SLU and dialog state correctness. After the evaluation, data inconsistencies were discovered in one of the test sets (Test 4, cf. Table 1). As result, that test set has been excluded from all results reported in this paper. Example dialogs from **DSTC1** are provided in the Appendix.

DSTC2 and **DSTC3** use a large corpus of dialogs with various telephone-based dialog systems that was collected using Amazon Mechanical Turk. The dialogs used in the challenges come from 6 conditions; all combinations of one of three possible dialog managers and one of two possible speech recognisers. There are roughly 500 dialogs in each condition, of average length 7.88 turns from 184 unique callers. More descriptive statistics are given in Table 1. Example dialogs from **DSTC2** and **DSTC3** are provided in the Appendix.

4.4 Dialog state definition and labeling

In **DSTC1**, the dialog state consists of a frame of **informable slots** which are slots provided by the user that describe their goal, such as the bus route and origin bus stop. The slots and approximate number of values for each are shown in Table 2. To determine the true dialog state, first each SLU hypothesis on each SLU N-Best list was manually labeled for its correctness. Each SLU hypothesis could contain values for more than one slot, such as `from=downtown,to=airport`. In making labeling decisions, the labeler could view the dialog history, and it was possible that zero, one, or more than one SLU hypothesis were labeled as correct. If the value for a slot had been provided but no correct value appeared in the SLU results, a special value called *rest* was considered to be correct.² At every turn, trackers output a scored list of values for every slot, including the special *rest* value. For evaluation, a dialog state was scored as correct if all of its slots were assigned values which had previously been marked as correct, or *rest* if no correct values had yet been observed for that slot value. Thus, in **DSTC1**, there could be multiple correct dialog states, and the best possible tracker could achieve 100% accuracy. Note that, in **DSTC1**, there was no explicit set of slot values,

2. The term *rest* refers to the remainder, as in “the rest of the unenumerated SLU hypotheses”.

		# Dialogs	Goal Changes	WER	SLU F-score
DSTC1	Train ¹	2,344	-	46.4%	45.3%
	Train+ ²	10,619	-	42.0%	-
	Test ³	2,485	-	55.1%	38.5%
DSTC2	Train	1,612	40.1%	26.4%	75.7%
	Devel.	506	37.0%	31.9%	71.6%
	Test	1,117	44.5%	28.7%	73.8%
DSTC3	Train ⁴	3,235	41.1%	28.1%	74.3%
	Test	2,275	16.5%	31.5%	78.1%

Table 1: Statistics for the data sets for all three challenges. *Goal Changes* is the percentage of dialogs in which the user changed their mind for at least one slot. *WER* and *SLU F-score* are on the top ASR and SLU hypotheses respectively. Further details of the datasets are given in Williams et al. (2013), Henderson et al. (2014b), and Henderson et al. (2014a).

¹This row combines sets Train 1A, Train 2 and Train 3 from **DSTC1**.

²This row combines sets Train 1B and 1C from **DSTC1**. In these dialogs, user speech was transcribed, but SLU and dialog state correctness were not labeled.

³This row combines sets Test 1, Test 2, and Test 3 from **DSTC1**. In this paper, Test 4 has been excluded due to data issues.

⁴The Training set for **DSTC3** is the combination of Train, Dev, and Test sets from **DSTC2**.

Slot	Size
bus route	100
date	-
time	-
origin street	500-10,000
origin neighborhood	20-100
origin PoI	50-500
destination street	500-10,000
destination neighborhood	20-100
destination PoI	500-10,000

Table 2: Slots used for **DSTC1** and their approximate number of values. The ranges of values are due to the fact that systems used to collect the dialogs had different internal designs and covered different numbers of street descriptions, neighborhoods and Points of Interests (PoI).

because dialogs were recorded from systems built by different research groups without a shared ontology.

Slot	DSTC2 Train	DSTC2 Test	DSTC3 Train	DSTC3 Test	Informable
type	1	1	1	3	yes
area	5	5	5	15	yes
food	91	91	91	28	yes
name	113	113	113	163	yes
pricerange	3	3	3	4	yes
near	—	—	—	52	yes
hastv	—	—	—	2	yes
hasinternet	—	—	—	2	yes
childrenallowed	—	—	—	2	yes
addr	—	—	—	—	no
phone	—	—	—	—	no
postcode	—	—	—	—	no

Table 3: Ontology used in **DSTC2** and **DSTC3** for tourist information. Counts do not include the special *Dontcare* value.

In **DSTC2-3**, the dialog state and labeling procedure was defined somewhat differently. In addition to **informable slots**, the dialog state in **DSTC2-3** included 2 other quantities. First, the state included **requested slots**, which are the slots the user wants to retrieve, such as the phone number, or price range (of a restaurant). Second, the state included the **search method** which indicated if the user wanted to query by providing constraints, providing the name of a restaurant, navigating a results list, etc. The values and sizes of all of the slots in **DSTC2-3** are given in Table 3. In **DSTC2-3**, informable slots could take a special value called *dontcare* which means the user said they had no preference for that slot – for example, “I don’t mind which type of food.”

In addition, **DSTC2-3** was based on an explicit ontology. Because of this, unlike in **DSTC1**, user requests in **DSTC2-3** were labeled with slot-value pairs taken from the ontology, regardless of the correctness of the SLU output. As a result, in **DSTC2-3**, at each turn there was a single correct dialog state, and because the SLU often did not contain the correct interpretation, a tracker that took the SLU as input could at best achieve less than 100% accuracy.

In addition to labeling dialog state, all user speech for all datasets was transcribed, either through crowd-sourcing or professional services.

4.5 Tracker output and evaluation metrics

Each tracker outputs a probability distribution over the set of possible dialog states. The goal is to assign probability 1.0 to the correct state, and 0.0 to other states. In each dialog state hypothesis output by a tracker, every slot is scored, so to be correct, the hypothesis must have perfect precision and recall.

Based on the ground truth, a number of metrics were computed on each tracker’s output. **Accuracy** measures the percent of turns where the top-ranked hypothesis is correct. This indicates the correctness of the item with the maximum score. **L2** measures the L^2 distance between the vector of scores, and a vector of zeros with 1 in the position of the correct hypothesis. This indicates the quality of all scores, when the scores are viewed as probabilities.

AvgP measures the mean score of the first correct hypothesis. This indicates the quality of the score assigned to the correct hypothesis, ignoring the distribution of scores to incorrect hypotheses. **MRR** measures the mean reciprocal rank of the first correct hypothesis. This indicates the quality of the ordering of the scores (without necessarily treating the scores as probabilities).

In addition, two versions of the receiver-operating characteristic (ROC) curves were computed, which measure the discrimination of the score for the highest-ranked state hypothesis. **ROC.V1** computes ROC as a fraction of all utterances, and **ROC.V2** computes fractions of correctly classified utterances. From each of these two curves, four values were extracted. **ROC.V1.EER** and **ROC.V2.EER** give the equal error rate – i.e., the value at which the number of false accepts and number of false rejects are equal. Using the V1 curve, **ROC.V1.CA05**, **ROC.V1.CA10**, **ROC.V1.CA20** give the percent of correctly accepted utterances when the false-accept rate is set to 5%, 10%, and 20%, respectively; and using the V2 curve, **ROC.V2.CA05**, **ROC.V2.CA10**, **ROC.V2.CA20** give the percent of correctly accepted utterances when the false-accept rate is set to 5%, 10%, and 20%, respectively.

In addition, several additional metrics were computed for **DSTC2-3**. **Neglogp** is the mean negative logarithm of the score given to the correct hypothesis, $-\log p_i$. Sometimes called the *negative log likelihood*, this is a standard score in machine learning tasks. Two metrics, **Update precision** and **Update accuracy** measure the accuracy and precision of updates to the top scoring hypothesis from one turn to the next. For more details, see Higashinaka et al. (2004), which finds these metrics to be highly correlated with dialog success in their data.

Apart from *what* to measure, *when* to measure – i.e., which turns to include when computing each metric, must also be defined. For **DSTC1**, a set of 3 *schedules* were used. **schedule1** includes every turn. **schedule2** include turns where the target slot is either present on the SLU n-best list, or where the target slot is included in a system confirmation action – i.e., where there is some observable new information about the target slot. **schedule3** includes only the last turn of a dialog. For **DSTC2** and **DSTC3**, user goals can change during a dialog, making **schedule3** less meaningful. Consequently, only **schedule1** and **schedule2** were used for these challenges.

4.6 Baselines

All three challenges featured a common simple baseline that mimics standard (non-statistical) approaches commonly used in spoken dialog systems, denoted ‘team0 entry0’. It maintains a single hypothesis for each slot. Its value is the SLU 1-best with the highest confidence score observed so far, with score equal to that SLU item’s confidence score. In addition, **DSTC1** featured a simpler majority baseline which always selects the *rest* hypothesis for each turn. Two more baselines were provided for **DSTC2** and **DSTC3**. The *focus* baseline, denoted ‘team0 entry1’, includes a simple model of changing goal constraints. Beliefs are updated for the goal constraint $s = v$, at turn t , $P(s = v)$, using the rule:

$$P(s = v)_t = q_t P(s = v)_{t-1} + SLU(s = v)_t \quad (2)$$

where $0 \leq SLU(s = v)_t \leq 1$ is the SLU confidence score for $s = v$ given by the SLU in turn t , and $q_t = \sum_{v'} SLU(s = v')_t \leq 1$.

Another baseline tracker, based on the tracker presented in Wang and Lemon (2013) is included in the evaluation, denoted ‘team0 entry2’. This tracker uses a selection of domain independent rules to update the beliefs, similar to the focus baseline. One rule uses a learnt parameter called

the noise adjustment, to adjust the SLU scores. Finally, an oracle tracker is included in **DSTC2** and **DSTC3** under the label ‘team0 entry3’. This reports the correct label with score 1 for each component of the dialog state, but only if it has been suggested in the dialog so far by the SLU. This gives an upper-bound for the performance of a tracker which uses only the SLU and its suggested hypotheses.

4.7 Participants

Participation to each challenge was free to any group willing to submit one or more entries by the challenge evaluation deadline. Participants were kept anonymous and only referred to in terms of team and entry numbers (e.g. team2.entry4), except when they chose to give their identity in their own published papers. Between 7 and 9 research groups participated in each challenge, fielding between 27 and 31 trackers in total, as shown in Table 4.

	# Teams	# Trackers
DSTC1	9	27
DSTC2	9	31
DSTC3	7	28

Table 4: Participation statistics for all three challenges. A subset of teams entered multiple **DSTCs**.

5. Challenge entries and results

5.1 Which metrics are appropriate for dialog state tracking?

As mentioned above, the evaluation in each of the **DSTCs** measured numerous properties of each entry, including accuracy, probability quality, score discrimination, etc. Therefore, the question immediately arises which metrics are most appropriate to study. Two studies have examined this question.

First, in **DSTC1**, metrics were clustered by their correlations with each other, and found to form 4 clusters: one related to correctness with **Accuracy**, **MRR**, and the three **ROC.V1.CA** metrics; a second related to probability quality with **L2** and **AvgP**; a third related to score discrimination with only **ROC.V1.EER**; and a fourth also related to score discrimination with the **ROC.V2.CA** measures (Williams et al., 2013). This study suggests that, within each cluster, it is sufficient to choose a single metric, since all metrics within a cluster will empirically yield nearly the same ordering of entries.

Second, in **DSTC2**, the question of what to measure was posed differently, as “Which evaluation metric and schedule would best predict improvement in overall dialog performance?” (Lee, 2014). The author uses the data to optimize a reinforcement learning-based dialog manager, then runs a regression analysis to see which metrics are the best predictors of end-to-end dialog performance. **L2**, **AvgP**, and **Accuracy** are found to be the most predictive. The study also finds that evaluating the joint goal is more predictive than evaluating slots in isolation, and that metrics evaluating only discrimination (e.g., **ROC.V2**) are not good predictors of dialog performance.

Given these findings, we focus on **Accuracy** and **L2** on joint goals throughout the results section. For consistency, we report results on **schedule2**. We note, however, that all metrics from every tracker in all three **DSTCs** are publicly available for analysis.

5.2 What were the entries, and what was their performance?

Tables 5-7 show the entries with the highest joint goal accuracy from each team in **DSTC1-3**, using **schedule 2**. The descriptions in these tables are based on a participant survey included with each of the **DSTCs**, and references are provided if teams identified their entry in a publication.

5.3 What types of errors do the trackers make?

As pointed out by Smith (2014), it is important to examine the types of errors made by a tracker in order to make improvements. To do this, at each turn, we compare the top dialog state output by each tracker with the true dialog state, and examine each slot. If a slot value is present in both the true and output dialog states and the slot values are equal, we mark the slot as *correct*. If the slot value is present in both the true and output dialog states and the slot values are *not* equal, we mark the slot as *wrong* – i.e., a substitution error. If the slot value is present in the true dialog state but not in the output dialog state, we mark the slot as *missing* – i.e., a deletion error. Finally, if the slot value is not present in the true dialog state but is present in the output dialog state, we mark the slot as *extra* – i.e., an insertion error. Note that, since there are multiple slots in a dialog state, a single turn may have multiple slot-level errors.

Results are given in Figure 3 (p. 22), including performance of the best baselines. These results show that the dominant error type is missing slots. Since all error types were scored equally, this result suggests that teams were rather conservative about guessing the slot value when confidence was low. It also suggests that recall in the upstream SLU is an important issue.

5.4 How much opportunity for improvement remains?

We next compared each tracker to the strongest baseline, and computed the percentage of turns where the tracker was correct and the baseline was not, and the percentage of turns where the baseline was correct and the tracker was not.

Results are shown in Figure 4. Even the best trackers – which *in total* make fewer errors than the baseline – still make some errors that the baselines do not. This implies that there is additional scope for improvement, perhaps through combining multiple trackers using ensemble methods (Lee and Eskenazi, 2013; Sun et al., 2014b; Henderson et al., 2014b).

5.5 What is the value beyond SLU?

Figure 5 shows the same analysis for an “SLU-based oracle tracker”, again for the best-performing entry for each team. This tracker considers the items on the SLU N -best list – it is an “oracle” in the sense that, if a slot/value pair appears that corresponds to the user’s goal, it is added to the state with confidence 1.0. In other words, when the user’s goal appears somewhere in the SLU N -best list, an oracle in **DSTC1** would always achieve perfect accuracy. The only errors made by the oracle are omissions of slot/value pairs which have not appeared on any SLU N -best list. Due to the use of the *rest* meta-value in **DSTC1**, the oracle always achieves 100% accuracy (c.f. Section 4.4). Therefore only results for **DSTC2** and **DSTC3** are shown.

Figure 5 shows that, for the best trackers, 5% or more of tracker turns outperformed the oracle. These teams also used ASR features, which indicates they were successfully using ASR results or dialog history to infer new slot/value pairs – i.e., to improve the recall of the existing SLU.

Unsurprisingly, despite these gains no team was able to achieve a net performance gain over the oracle.

5.6 What is the state-of-the-art?

Synthesizing the results above, we can summarize the properties of state-of-the-art dialog state trackers:

- **Discriminative models:** The strongest entries are consistently discriminative models. Although some rule-based systems have achieved notable performance – for example, team2 entry1 in **DSTC1**, team3 entry1 in **DSTC2**, and team4 entry0 in **DSTC3** – in no case has a rule-base or generative model achieved best performance in any of the **DSTCs**.
- **Use ASR features:** The best trackers consistently incorporate low-level ASR features. Low-level ASR features such as N-best scores and word confusion network scores provide additional signals that improves precision (Williams, 2014). Further, incorporating the ASR results themselves yields additional dialog state hypotheses that improve recall (Section 5.5).
- **Sequential:** The best trackers either model dialog directly as a sequence – CRFs for team6 entry4 in **DSTC1** and RNNs for team4 in **DSTC2** and team3 in **DSTC3** – or otherwise incorporate extensive dialog history features, as in team2 entries 1 and 3 in **DSTC3**, which used hundreds of features from the dialog history. Passing only the distribution over hidden states from one turn to the next, as is often done with generative or rule-based approaches, does not perform as well. Relying on the distribution over states assumes that state transitions are Markovian; this result suggests that states may be encoding insufficient history for the Markov assumption to be valid.
- **Capture feature interactions:** The best trackers directly model interactions between features. For example, the best trackers in **DSTC2** and **DSTC3** directly modeled feature interactions, either via (recurrent) neural networks or collections of decision trees. Approaches that do not capture feature interactions, such as log-linear models where each feature of a dialog state affects its score independently – for example, team5 entry1 in **DSTC1** – were not top finishers.
- **Joint posteriors:** In **DSTC1** and **DSTC2**, the best systems computed a joint posterior over all slots, rather than computing a posterior as a product of the marginals for each slot. The gain observed in **DSTC1** (Table 5b) was particularly large, whereas the gain in **DSTC2** was present but small (Henderson et al., 2014b). This difference is probably due to differences in the domains: bus stops and bus routes requested by real callers in **DSTC1** were highly correlated, whereas the subjects in **DSTC2** and **DSTC3** were given a specification with slot values drawn closer to uniform.

6. Practical issues and lessons learned

The main effort in organizing the DSTC series was the preparation of the data. In **DSTC1**, this task was particularly labor-intensive because there was no ontology of bus stops available, which

Entry	Reference	Description
team1 entry1	(Henderson et al., 2013)	Deep neural network
team2 entry1	(Wang and Lemon, 2013)	Hand-crafted rules based on confidence scores
team3 entry2	(Zilka et al., 2013)	Discriminative classifier + hand-crafted transition probabilities
team4 entry1	(anonymous)	Discriminative dynamic Bayesian network
team5 entry1	(Williams, 2013)	Decision tree
team6 entry4	(Lee and Eskenazi, 2013)	Discriminative + generative (system combination); unsupervised prior adaptation
team7 entry1	(anonymous)	Discriminatively trained graphical model
team8 entry4	(anonymous)	Support vector machines
team9 entry4	(Kim et al., 2013)	Generative plus discriminative re-scoring.

(a) **DSTC1** entries. References cited where teams identified their entry in a published paper. Description based on survey collected from participants.

Entry	Features		Goals		Joint Goals	
	ASR	SLU	Acc.	L2	Acc.	L2
majority class baseline ¹		✓	0.554	0.631	0.166	1.180
1-best baseline ¹		✓	0.564	0.599	0.241	1.078
team1 entry1		✓	0.674	0.612	0.349	1.067
team2 entry1		✓	0.683	0.532	0.354	1.055
team3 entry2		✓	0.650	0.503	0.339	0.964
team4 entry1		✓	0.565	0.626	0.278	1.045
team5 entry1		✓	0.691	0.503	0.237	1.087
team6 entry4		✓	0.765	0.443	0.466	0.890
team7 entry1		✓	0.615	0.562	0.283	1.058
team8 entry4		✓	0.584	0.592	0.226	1.098
team9 entry4		✓	0.724	0.492	0.357	1.024
SLU-based oracle		✓	1.000	0.000	1.000	0.000

(b) **DSTC1** results. The top performing trackers from each team are selected. Results are derived from combining all test sets in the evaluation. In **DSTC1**, none of the entries used the ASR output. ¹Williams et al. (2013).

Table 5: Entries and results of **DSTC1**.

Entry	Reference	Description
team1 entry0	(Kim and Banchs, 2014)	Linear CRF
team3 entry0	(Smith, 2014)	Discourse rules + dialog act bigrams
team4 entry2	(Henderson et al., 2014d)	Recurrent neural network
team6 entry2	(anonymous)	Maximum entropy Markov model, with DNN output distribution
team7 entry4	(Sun et al., 2014b)	System combination of a Deep neural network and maximum entropy model
team8 entry1	(Lee et al., 2014)	Hidden Information State Model + Goal Change Handling Model + System-User Action Pair weighting Model
team9 entry0	(anonymous)	Baseline, augmented with priors from a confusion matrix
team2 entry2	(Williams, 2014)	Recurrent neural network
team4 entry0	(Henderson et al., 2014d)	Recurrent neural network
team7 entry0	(Sun et al., 2014b)	System combination of a Deep neural network, maximum entropy model, and rules
team2 entry1	(Williams, 2014)	Ranking (lambdaMART)
team2 entry3	(Williams, 2014)	Ranking (lambdaMART)
team5 entry4	(anonymous)	ASR/SLU re-ranking

(a) **DSTC2** entries. References cited where teams identified their entry in a published paper. Description based on survey collected from participants.

Entry	Features		Joint Goals		Search Method		Requested	
	ASR	SLU	Acc.	L2	Acc.	L2	Acc.	L2
1-best baseline ¹		✓	0.619	0.738	0.879	0.209	0.884	0.196
focus baseline ¹		✓	0.719	0.464	0.867	0.210	0.879	0.206
HWU baseline ²		✓	0.711	0.466	0.897	0.158	0.884	0.201
team1 entry0		✓	0.601	0.648	0.904	0.155	0.960	0.073
team3 entry0		✓	0.729	0.452	0.878	0.210	0.889	0.188
team4 entry2		✓	0.742	0.387	0.922	0.124	0.957	0.069
team6 entry2		✓	0.718	0.437	0.871	0.210	0.951	0.085
team7 entry4		✓	0.735	0.433	0.910	0.140	0.946	0.089
team8 entry1		✓	0.699	0.498	0.899	0.153	0.939	0.101
team9 entry0		✓	0.499	0.760	0.857	0.229	0.905	0.149
team2 entry2	✓		0.668	0.505	0.944	0.095	0.972	0.043
team4 entry0	✓		0.768	0.346	0.940	0.095	0.978	0.035
team7 entry0	✓		0.750	0.416	0.936	0.105	0.970	0.056
team2 entry1	✓	✓	0.784	0.735	0.947	0.087	0.957	0.068
team2 entry3	✓	✓	0.771	0.354	0.947	0.087	0.941	0.090
team5 entry4	✓	✓	0.695	0.610	0.927	0.147	0.974	0.053
SLU-based oracle ¹		✓	0.850	0.300	0.986	0.028	0.957	0.086

(b) Results of **DSTC2** evaluation. The top performing trackers from each team are selected. Results are split by the input features used. ¹Henderson et al. (2014b), ²Wang and Lemon (2013).

Table 6: Entries and results of **DSTC2**.

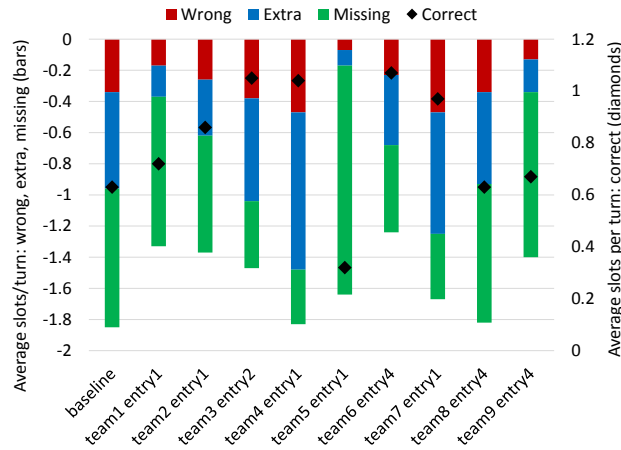
Entry	Reference	Description
team1 entry3	(anonymous)	Rules with parameters inferred from data
team6 entry0	(anonymous)	Generative model trained with cascading gradient descent
team7 entry1	(Ren et al., 2014a)	Markovian neural network model
team3 entry2	(Henderson et al., 2014c)	Recurrent neural network
team5 entry0	(Sun et al., 2014a)	Rules that operate on confidence scores
team2 entry0	(anonymous)	Maximum entropy model
team2 entry3	(anonymous)	System combination: maximum entropy, CRF, rules
team3 entry0	(Henderson et al., 2014c)	Recurrent neural network
team4 entry0	(Kadlec et al., 2014)	Rules with parameters inferred from data

(a) **DSTC3** entries. References cited where teams identified their entry in a published paper. Description based on survey collected from participants.

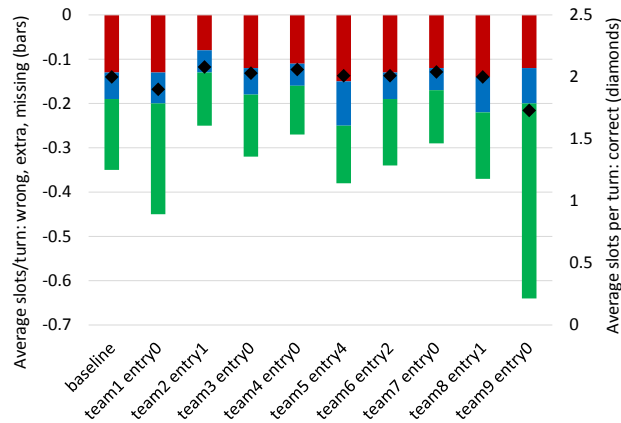
	Features		Joint Goals		Search Method		Requested	
	ASR	SLU	Acc.	L2	Acc.	L2	Acc.	L2
1-best baseline ¹		✓	0.555	0.860	0.922	0.154	0.778	0.393
focus baseline ¹		✓	0.556	0.750	0.908	0.134	0.761	0.435
HWU baseline ²		✓	0.575	0.744	0.967	0.062	0.767	0.417
team1 entry3		✓	0.561	0.733	0.963	0.097	0.774	0.401
team6 entry0		✓	0.507	0.736	0.927	0.120	0.907	0.157
team7 entry1		✓	0.576	0.652	0.957	0.116	0.938	0.101
team3 entry2	✓		0.616	0.565	0.966	0.061	0.939	0.100
team5 entry0	✓		0.610	0.556	0.968	0.091	0.949	0.090
team2 entry0	✓	✓	0.585	0.697	0.965	0.114	0.929	0.121
team2 entry3	✓	✓	0.582	0.639	0.970	0.065	0.938	0.138
team3 entry0	✓	✓	0.646	0.534	0.966	0.061	0.943	0.091
team4 entry0	✓	✓	0.630	0.627	0.853	0.272	0.923	0.136
SLU-based oracle ¹		✓	0.717	0.565	0.988	0.02	0.946	0.107

(b) Results of **DSTC3** evaluation. The top performing trackers from each team are selected. Results are split by the input features used, with bold indicating the top result in the group. ¹Henderson et al. (2014a), ²Wang and Lemon (2013).

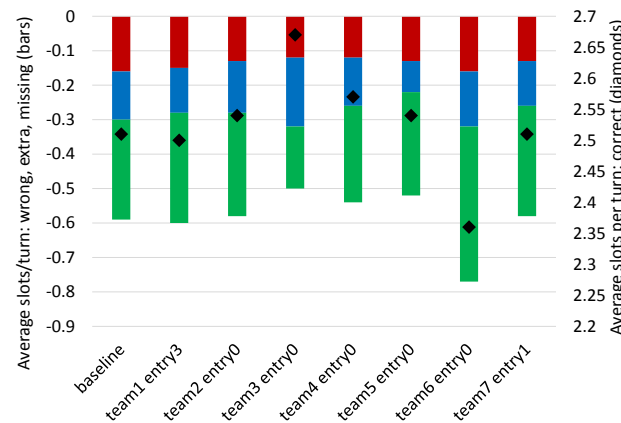
Table 7: Entries and results of **DSTC3**.



(a) DSTC1



(b) DSTC2



(c) DSTC3

Figure 3: Average number of slots in error per turn (bar chart, left axis), and average number of correct slots per turn (black diamonds, right axis) for the best tracker from each team in each of the **DSTCs**. See text for explanation of error types. The left axis shows negative numbers so that the top of each plot indicates ideal performance for both errors (bars) and correctness (diamonds). Team IDs are not consistent across different **DSTCs**.

DIALOG STATE TRACKING OVERVIEW

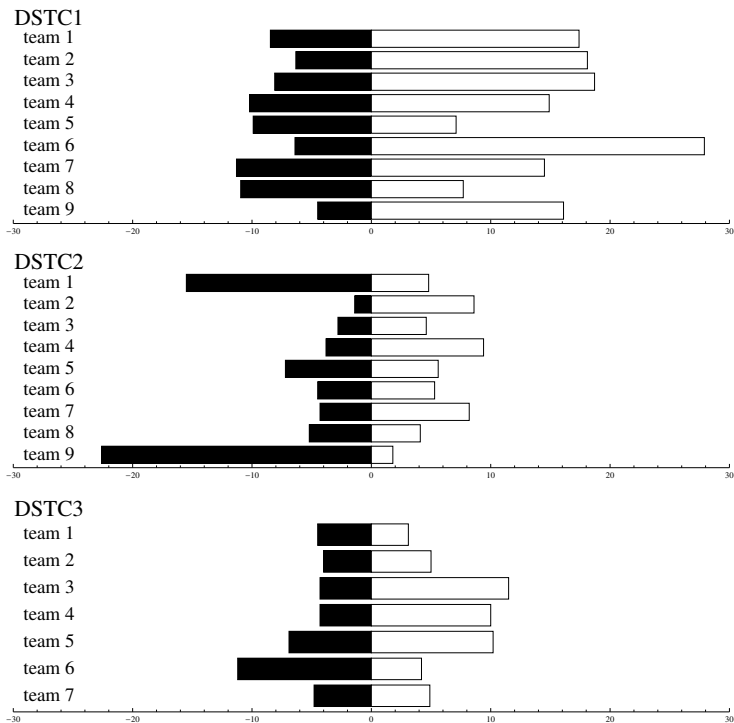


Figure 4: Percentage of all turns where the top tracker from each team did better than the baseline (white bar) or worse than the baseline (black bar) for the joint goal accuracy metric. Note that the team IDs are not consistent across different **DSTCs**.

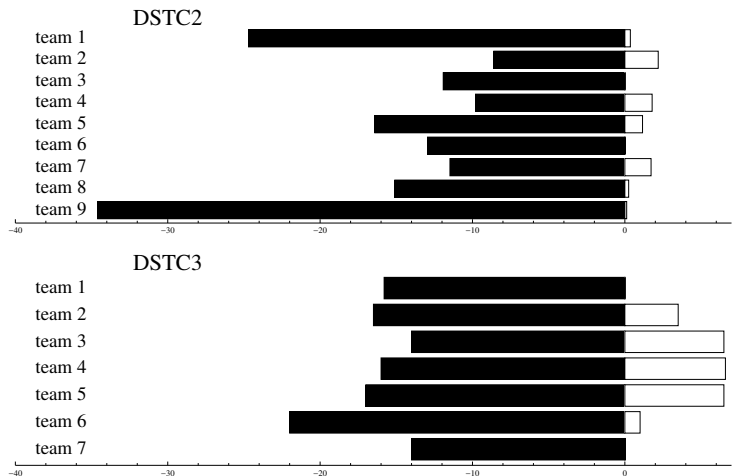


Figure 5: Percentage of all turns where the top tracker from each team did better than the SLU-based oracle (white bar) or worse than the oracle (black bar) for the joint goal accuracy metric. **DSTC1** is not shown because its design resulted in the oracle always achieving 100% accuracy, so it was not possible to beat the performance of the oracle in **DSTC1**. Note that the team IDs are not consistent across different **DSTCs**.

required manually labeling each SLU hypothesis for correctness. This was done by a mixture of professional transcribers and crowd workers, at a cost of a few thousand dollars. Edge cases were difficult for either group, and many utterances needed to be manually labeled by the organizers, often by consulting native Pittsburghers or researching Pittsburgh geography. A further difficulty in preparing the data for **DSTC1** was the need to design a dialog act ontology that represented dialog acts produced by three dialog systems from different research groups. By comparison, preparing the data for **DSTC2-3** required much less work because an explicit ontology simplified labeling, and dialogs were drawn from a single system which included system-side dialog act tags. Future challenges might benefit from following the approach in **DSTC2-3**.

The DSTC organizers decided to continue to make the data freely available after the conclusion of the challenge. This has had unforeseen benefits: first, the DSTC data now forms a sort of benchmark for the field, with groups continuing to report results on it after the challenge proper (Lee, 2013; Ma and Fosler-Lussier, 2014b; Zilka and Jurčiček, 2015; Fix and Frezza-Buet, 2015). In addition, the **DSTC1-3** corpora have been used to examine which state tracking evaluation metrics correlate with dialog success (Lee, 2014), perform detailed error analyses of state trackers (Smith, 2014), and for dialog act classification and SLU experimentation (Ma and Fosler-Lussier, 2014a; Ferreira et al., 2015). We encourage future challenges to continue this tradition.

7. Perspectives and Conclusion

Although dialog state tracking is a crucial problem in spoken dialog systems, until recently it received only sporadic attention. Throughout the 1990s, hand-crafted rules were the dominant solution in both research and production systems. In the early 2000s, researchers recognized the need to model uncertainty explicitly and make use of all of the information on the SLU N-Best list, and proposed several methods, with generative models being most common. Yet work was sporadic and different methods were rarely compared: different groups operated their own dialog systems, and there was no standardized dataset and framework for evaluation.

The Dialog State Tracking Challenge has introduced the first shared datasets and common evaluation metrics for this problem, and has catalyzed substantial new work into this research problem. In particular, the DSTC series has underpinned three broad advances.

The first contribution of the DSTC series has been to change the dominant approach from generative models to discriminatively trained classifiers. Prior to the DSTC series, generative models were most common. The DSTC series has illustrated the weaknesses in generative models that hindered accuracy, such as the inability to handle a large number of features. In their simplest form, discriminatively trained classifiers take as input a feature vector of fixed size, where the features summarize dialog history up to the current turn.

The second contribution of the DSTC series has been to enable the development of discriminative sequential models for dialog state tracking. Unlike simple classifiers, sequential models take as input a set of features at *each turn*, avoiding the need to design features that summarize the dialog history. Thus, sequential models substantially simplify the feature engineering process, reducing effort. Because they properly account for dialog as a temporal process, they also have the potential to improve accuracy, and this has been demonstrated in DSTC entries.

The third and most recent contribution of the DSTC series has been to underpin models which take the ASR results as input directly, eschewing the SLU entirely. This move further reduced the feature engineering effort – these methods use only primitive ASR features and require essentially no feature design at all. By providing direct access to the raw input signal, they also have the

potential to provide a further improvement in accuracy, which has also been demonstrated in the DSTC series.

A key outstanding question for the field is whether improvements in dialog state tracking performance translate to improvements in end-to-end dialog system performance, such as improved task completion or user satisfaction. Two early studies show promising results. First, Lee et al. (2014) performed off-line reinforcement learning experiments on the (static) **DSTC1** corpus, and showed that improved dialog state tracking performance is indeed correlated with improved dialog performance. Second, Kim et al. (2014) constructed a user simulator, and used simulated dialogs to compare an existing generative tracker with a discriminative tracker that had been shown to yield better dialog state tracking accuracy. They found that the discriminative tracker yielded better end-to-end dialog performance. The use of a simulated user and learned dialog policy implies that the distribution of dialog states encountered at test time was different than that encountered in training, and the fact that the discriminative tracker performed well suggests it had not overfit the training data. While these are promising results, the definitive test will be a deployment with real users, which (to our knowledge) has not yet been done. We encourage future DSTC instances to consider end-to-end evaluations with real users.

The first three instances of the dialog state tracking challenge have tackled human-computer interaction in two domains with dialog systems from multiple research groups, including situations where users' goals may change and where new slots are present at runtime. Given the positive results achieved to date, the next instance of the challenge – **DSTC4** – will tackle state tracking in *human-human* conversation (Kim et al., 2016). Human-human conversation presents a host of new challenges, and addressing these is a step toward the long-standing goal of learning a dialog system from example human-human conversations.

8. Acknowledgements

The DSTC series has benefited from the encouragement and support of many of our fellow researchers. First, the authors thank Maxine Eskenazi, Alan Black, and David Traum for organizing the 2012 NAACL workshop on “Future directions and needs in the Spoken Dialog Community: Tools and Data”, which catalysed the first challenge (Williams, 2012b). Thanks also to the DSTC advisory boards for their guidance and input in all three challenge tasks. The authors thank SIGDIAL for their endorsement, and to SIGDIAL 2013 and 2014, and IEEE SLT 2014 for hosting special sessions. Thanks to AT&T, Carnegie Mellon University, Microsoft, and Cambridge University for providing data and labels, and to Microsoft and Honda Research Institute for sponsorship to fund additional labeling. Thanks to our co-organizers, Deepak Ramachandran, Alan Black, and Blaise Thomson. Finally, a special thank you to all of the DSTC participants.

Appendix 1: Sample dialog excerpts

Below are transcripts of calls in the DSTC corpus. The user's side shows utterances as transcribed by human annotators.

DSTC1 : Group A

SYS: Welcome to the CMU Let's Go bus information system. To get help at any time, just say Help or press zero. What can I do for you?

USR: *61c evening hours*

SYS: For example, you can say, when is the next 28X from DOWNTOWN to THE AIRPORT? or I'd like to go from MCKEESPORT to HOMESTEAD tomorrow at 10 a.m.

USR: *i'd like to go [noise]*

SYS: Where are you leaving from?

USR: *from Kennywood tomorrow at 6:30 PM*

SYS: Leaving from MCKEE. Is this correct?

USR: *yes*

SYS: Okay What is your destination?

USR: *downtown*

SYS: Going to DOWNTOWN. Is this correct?

USR: *yes*

SYS: Right When do you want to travel?

USR: *6:30 PM*

SYS: Leaving at 6 30 p.m.. Is this correct?

USR: *[noise] yes*

SYS: Alright Hold on. I'll look that up. There is a 71D leaving FIFTH AVENUE AT DARRAGH MCKEE at 6 40 p.m.. It will arrive at FIFTH AVENUE AT MARKET at 6 58 p.m.. To get more information about buses related to this trip, you can say, when is the next bus, or, when is the previous bus. To ask about a different trip, you can say, start a new query. If you are finished, you can say goodbye.

USR: *start a new query*

SYS: Okay, let's start from the beginning.

USR: ...

DSTC1 : Group B

SYS: East Pittsburgh Bus Schedules. Say a bus route, like 28X, or say I'm not sure.

USR: *61A*

SYS: Okay, 61A. To change, say go back. Where are you leaving from?

USR: *Downtown*

SYS: Okay, downtown. You can always say go back. And where are you going to?

USR: *East Pittsburgh East Pittsburgh*

SYS: Pittsburgh. Is that right?

USR: *No*

SYS: Oh, sorry, where are you going to?

USR: *East Pittsburgh*

SYS: Okay, East Pittsburgh. Do you want times for the next few buses? Say yes or no.

USR: *Yes*

SYS: Ok, the next few buses. To change, say go back. At 7:45 PM today, there is a 61 A, from Fifth Avenue And Market Street, Downtown, arriving Dynamo Alley And Beech Avenue, In East Pittsburgh, at 8:42 PM. Say repeat, next, previous, or start over.

DSTC1 : Group C

SYS: Welcome to the Lets Go! bus information system for Pittsburgh. Where would you like to leave from?

USR: *[noise]*

SYS: Where do you wanna leave from?

USR: *[noise] um I'm leaving from Forbes avenue and Murray avenue*

SYS: I'm sorry but I do not know any stops in the moon neighborhood. The routes I currently cover are the following: 28X, 54C, 56U, 61A, 61B, 61C, 61D and 64.

USR: *um I need the scheduling information for the 61D*

SYS: Where would you like to leave from?

USR: *Forbes and Murray*

SYS: Where do you wanna go?

USR: *Greenfield*

SYS: When are you going to take that bus?

USR: *61D*

SYS: When are you going to take that bus?

USR: *61D*

SYS: When would you like to travel? Say next bus for the the next bus .

USR: *next bus*

SYS: The next bus from forbes avenue at murray to greenfield is the 61D, leaving at 8 43 p.m. . It arrives at murray avenue at loreta at 8 52 p.m. . You may now say. next bus, previous bus, restart, or goodbye.

USR: *[noise] good-bye*

SYS: Thank you goodbye

DSTC2 and DSTC3 (Training Set)

SYS: Which part of town?

USR: *The north uh area*

SYS: Which part of town?

USR: *A cheap place in the north*

SYS: Clown cafe is a cheap restaurant in the north part of town.

USR: *Do you have any others like that?*

SYS: Galleria is another cheap restaurant in the north.

USR: *What is their phone number and address?*

DSTC3 (Test Set)

SYS: Thank you for calling the Cambridge Information system. Your call will be recorded for research purposes. You may ask for information about a place to eat, such as a restaurant, a pub, or a cafe. How may I help you?

USR: *i would like to find a pub with an internet connection and a tv*

SYS: The Bakers, which has an internet connection, is a great pub with a television.

USR: *what is the address phone number and price range*

SYS: The Bakers is located on 176 East Road and its phone number is 01223304121. The price for main courses ranges from 6 pounds to 8 pounds 49 and the price for a cheapest pint of beer is 2 pounds 25.

USR: *[noise]*

SYS: The Bakers is an outstanding pub.

USR: *what is the address*

SYS: The Bakers is on 176 East Road.

USR: *what is the phone number*

SYS: The phone number of The Bakers is 01223304121.

USR: *what is the price range*

SYS: The Bakers is in the cheap price range.

USR: *thank you good bye*

References

- Alan Black, Susanne Burger, Brian Langner, Gabriel Parent, and Maxine Eskenazi. Spoken dialog challenge 2010. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), Berkeley, California, USA, 2010*.
- Dan Bohus and Eric Horvitz. Models for multiparty engagement in open-world dialog. In *Proc SIGdial Conf on Discourse and Dialogue, London, USA, 2009*.
- Dan Bohus and Alex Rudnicky. Ravenclaw: Dialogue management using hierarchical task decomposition and an expectation agenda. In *Proc Eurospeech, Geneva, Switzerland, 2003*.
- Dan Bohus and Alex Rudnicky. A ‘K hypotheses + other’ belief updating model. In *Proc AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems, Boston, USA, 2006*.
- Trung Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable hybrid DDN-POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Nat. Lang. Eng.*, 15(2):273–307, 2009.
- Herbert H Clark. *Using Language*. Cambridge University Press, 1996. ISBN 9780521567459.
- Philip R Cohen and Hector J Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 221–255. MIT Press, Cambridge, MA, 1990.
- David DeVault. *Contribution Tracking: Participating in Task-Oriented Dialogue under Uncertainty*. PhD thesis, Rutgers, The State University of New Jersey, 2008.
- David DeVault and Matthew Stone. Managing ambiguities across utterances in dialogue. In *Proc Workshop on the Semantics and Pragmatics of Dialogue (DECALOG), Trento, Italy, 2007*.
- Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefvre. Online adaptive zero-shot learning spoken language understanding using word-embedding. In *Proc Intl Conf on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015*.

- Jeremy Fix and Herve Frezza-Buet. Yabus : Yet another rule based belief update system. arXiv:1507.06837v1 [cs.CL], 2015.
- Milica Gasic and Steve Young. Effective handling of dialogue state in the hidden information state pomdp dialogue manager. *ACM Transactions on Speech and Language Processing*, 7, 2011.
- David Heckerman and Eric Horwitz. Inferring Informational Goals from Free-text Queries: a Bayesian Approach. In *Proc 14th Conf on Uncertainty in Artificial Intelligence (UAI)*, pages 230–238, 1998.
- James Henderson and Oliver 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, USA, 2008*.
- Matthew Henderson, Blaise Thomson, and Steve Young. Deep Neural Network Approach for the Dialog State Tracking Challenge. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France, 2013*.
- Matthew Henderson, Blaise Thomson, and Jason D. Williams. The third dialog state tracking challenge. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014a*.
- Matthew Henderson, Blaise Thomson, and Jason D Williams. The Second Dialog State Tracking Challenge. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA, 2014b*.
- Matthew Henderson, Blaise Thomson, and Steve Young. Robust Dialog State Tracking Using Delexicalised Recurrent Neural Networks and Unsupervised Adaptation. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014c*.
- Matthew Henderson, Blaise Thomson, and Steve Young. Word-Based Dialog State Tracking with Recurrent Neural Networks. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA, 2014d*.
- Matthew Henderson, Blaise Thomson, and Steve Young. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014e*.
- Ryuichiro Higashinaka, Mikio Nakano, and Kiyooki Aikawa. Corpus-based discourse understanding in spoken dialogue systems. In *Proc Association for Computational Linguistics (ACL), Sapporo, Japan, 2003*.
- Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyooki Aikawa. Evaluating discourse understanding in spoken dialogue systems. *ACM Trans. Speech Lang. Process.*, 2004.
- Eric Horvitz and Tim Paek. A Computational Architecture for Conversation. In *Proceedings of the 7th Intl Conf on User Modeling*, pages 201–210, Banff, Canada, 1999.
- Filip Jurčićek, Blaise Thomson, and Steve Young. Natural actor and belief critic: Reinforcement algorithm for learning parameters of dialogue systems modelled as POMDPs. *TSLP*, 7, 2011.

- Rudolf Kadlec, Miroslav Vodolan, Jindrich Libovicky, Jan Macek, and Jan Kleindienst. Knowledge-based dialog state tracking. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA*, 2014.
- Daejoong Kim, Jaedeug Choi Choi, Kee-Eung Kim, Jungsu Lee, and Jinho Sohn. Engineering statistical dialog state trackers: A case study on dstc. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, pages 462–466, 2013.
- Dongho Kim, Matthew Henderson, Milica Gasic, Pirros Tsiakoulis, and Steve Young. The use of discriminative belief tracking in pomdp-based dialogue systems. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA*, 2014.
- Kyungduk Kim, Cheongjae Lee, Sangkeun Jung, and Gary Geunbae Lee. A frame-based probabilistic framework for spoken dialog management using dialog examples. In *Proc SIGdial Workshop on Discourse and Dialogue, Columbus, Ohio, USA*, 2008.
- Seokhwan Kim and Rafael E. Banchs. Sequential labeling for tracking dynamic dialog states. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, 2014.
- Seokhwan Kim, Luis Fernando DHaro, Rafael E. Banchs, Jason D. Williams, and Matthew Henderson. The fourth dialog state tracking challenge. In *Proc Intl Workshop on Spoken Dialog Systems (IWSDS), Saariselka, Finland*, 2016.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc Intl Conf on Machine Learning (ICML), Massachusetts, USA*, pages 282–289, 2001.
- Staffan Larsson and David Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 5(3/4):323–340, 2000.
- Byung-Jun Lee, Woosang Lim, Daejoong Kim, and Kee-Eung Kim. Optimizing generative dialog state tracker via cascading gradient descent. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, pages 273–281, 2014.
- Sungjin Lee. Structured Discriminative Model For Dialog State Tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, 2013.
- Sungjin Lee. Extrinsic evaluation of dialog state tracking and predictive metrics for dialog policy optimization. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, 2014.
- Sungjin Lee and Maxine Eskenazi. Recipe For Building Robust Spoken Dialog State Trackers: Dialog State Tracking Challenge System Description. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, 2013.
- Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proc SIGdial Conf on Discourse and Dialogue, Prague, Czech Republic*, 2015.
- Yi Ma and Eric Fosler-Lussier. Detecting ‘request alternatives’ user dialog acts from dialog context. In *Proc Intl Workshop on Spoken Dialog Systems (IWSDS), Napa, California, USA*, 2014a.

- Yi Ma and Eric Fosler-Lussier. A discriminative sequence model for dialog state tracking using user goal change detection. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014b*.
- Yi Ma and Eric Fosler-Lussier. A discriminative sequence model for dialog state tracking using user goal change detection. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014c*.
- Yi Ma, Antoine Raux, Deepak Ramachandran, and Rakesh Gupta. Landmark-based location belief tracking in a spoken dialog system. In *Proc SigDial Conf on Discourse and Dialogue, Seoul, Korea, 2012*.
- Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, 2000.
- Neville Mehta, Rakesh Gupta, Antoine Raux, Deepak Ramachandran, and Stefan Krawczyk. Probabilistic ontology trees for belief tracking in dialog systems. In *Proc SigDial Conf on Discourse and Dialogue, Tokyo, Japan, 2010*.
- Helen Meng, Carmen Wai, and Roberto Pieraccini. The use of belief networks for Mixed-Initiative dialog modeling. *IEEE Transactions on Speech and Audio Processing*, 11(6):757–773, November 2003.
- Angeliki Metallinou, Dan Bohus, and Jason D Williams. Discriminative state tracking for spoken dialog systems. In *Proc Association for Computational Linguistics (ACL), Sofia, Bulgaria, 2013*.
- Tim Paek and Eric Horvitz. Conversation as action under uncertainty. In *Proc Conf on Uncertainty in Artificial Intelligence (UAI), Stanford, California, USA, pages 455–464, 2000*.
- Stephen Pulman. Conversational Games, Belief Revision and Bayesian Networks. In *CLIN VII: 7th Computational Linguistics in the Netherlands meeting, 1996*.
- Antoine Raux and Yi Ma. Efficient probabilistic tracking of user goal and dialog history for spoken dialog systems. In *Proc INTERSPEECH Conf, Florence, Italy, pages 801–804. ISCA, 2011*.
- Hang Ren, Weiqun Xu, Yan Zhang, and Yonghong Yan. Dialog State Tracking using Conditional Random Fields. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France, 2013*.
- Hang Ren, Weiqun Xu, and Yonghong Yan. Markovian discriminative modeling for cross-domain dialog state tracking. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA, 2014a*.
- Hang Ren, Weiqun Xu, and Yonghong Yan. Markovian discriminative modeling for dialog state tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA, pages 327–331, 2014b*.
- Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proc Conf on Empirical Methods in Natural Language Processing (EMNLP), Edinburgh, United Kingdom, 2011*.

- Nicholas Roy, Joelle Pineau, and Sebastian Thrun. Spoken dialog management for robots. In *Proc Association for Computational Linguistics (ACL), Hong Kong*, pages 93–100, 2000.
- Ronnie Smith. Comparative Error Analysis of Dialog State Tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, 2014.
- Kai Sun, Lu Chen, Su Zhu, and Kai Yu. A generalized rule based tracker for dialogue state tracking. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), South Lake Tahoe, Nevada, USA*, 2014a.
- Kai Sun, Lu Chen, Su Zhu, and Kai Yu. The SJTU system for dialog state tracking challenge 2. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, pages 318–326, 2014b.
- Umar Syed and Jason D. Williams. Using automatically transcribed dialogs to learn user models in a spoken dialog system. In *Proc Association for Computational Linguistics Human Language Technologies (ACL-HLT), Columbus, Ohio, USA*, pages 121–124, 2008.
- Blaise Thomson and Steve Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588, 2010.
- Blaise Thomson, Filip Jurčiček, Milica Gasic, Simon Keizer, François Mairesse, Kai Yu, and Steve Young. Parameter learning for POMDP spoken dialogue models. In *Proc IEEE Workshop on Spoken Language Technologies (SLT), Berkeley, California, USA*, 2010.
- Zhuoran Wang and Oliver Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, 2013.
- Jason D. Williams. Using particle filters to track dialogue state. In *Proc IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Kyoto, Japan*, 2007.
- Jason D. Williams. Exploiting the asr n-best by tracking multiple dialog state hypotheses. In *Proc INTERSPEECH Conf, Brisbane, Australia*, 2008.
- Jason D. Williams. Incremental partition recombination for efficient tracking of multiple dialogue states. In *Proc Intl Conf on Acoustics, Speech and Signal Processing (ICASSP), Dallas, Texas, USA*, 2010.
- Jason D. Williams. Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Advances in Spoken Dialogue Systems and Mobile Interface*, 6(8):959–970, 2012a.
- Jason D. Williams. A belief tracking challenge task for spoken dialog systems. In *NAACL HLT 2012 Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data, Montreal, Canada*, 2012b.
- Jason D. Williams. Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments. *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Advances in Spoken Dialogue Systems and Mobile Interface*, 6(8):959–970, 2012c.

- Jason D. Williams. Multi-domain learning and generalization in dialog state tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, 2013.
- Jason D. Williams. Web-style ranking and SLU combination for dialog state tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Philadelphia, USA*, 2014.
- Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.
- Jason D. Williams, Pascal Poupart, and Steve Young. Factored partially observable Markov decision processes for dialogue management. In *Proc Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Intl Joint Conf on Artificial Intelligence (IJCAI), Edinburgh, United Kingdom*, 2005.
- Jason D Williams, Antoine Raux, Deepak Ramachadran, and Alan Black. The Dialog State Tracking Challenge. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, August 2013.
- Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The Hidden Information State Approach to Dialog Management. In *ICASSP 2007, Honolulu, Hawaii*, 2007.
- Steve Young, Milica Gasic, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, 2010.
- Steve Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. POMDP-based Statistical Spoken Dialogue Systems: a Review. *Proceedings of the IEEE*, PP(99):1–20, 2013.
- Steve Young, Catherine Breslin, Milica Gasic, Matthew Henderson, Dongho Kim, Martin Szummer, Blaise Thomson, Pirros Tsiakoulis, and Eli Tzirkel Hancock. Evaluation of Statistical POMDP-based Dialogue Systems in Noisy Environment. In *Proc Intl Workshop on Spoken Dialog Systems (IWSDS), Napa, California, USA*, 2014.
- Bo Zhang, Qingsheng Cai, Jianfeng Mao, Eric Chang, and Baining Guo. Spoken dialogue management as planning and acting under uncertainty. In *Proc Eurospeech, Aalborg, Denmark*, pages 2169–2172, 2001.
- Lukas Zilka and Filip Jurčiček. Incremental LSTM-based dialog state tracker. arXiv:1507.03471v1 [cs.CL], 2015.
- Lukas Zilka, David Marek, Matej Korvas, and Filip Jurčiček. Comparison of bayesian discriminative and generative models for dialogue state tracking. In *Proc SIGdial Conf on Discourse and Dialogue, Metz, France*, 2013.
- Victor Zue, Stephanie Seneff, James R Glass, Joseph Polifroni, Christine Pao, Timothy J Hazen, and Lee Hetherington. Juplter: a telephone-based conversational interface for weather information. *Speech and Audio Processing, IEEE Transactions on*, 8(1):85–96, 2000.