User Goal Change Model for Spoken Dialog State Tracking

Yi Ma
Department of Computer Science & Engineering
The Ohio State University
Columbus, OH 43210, USA
may@cse.ohio-state.edu

Abstract

In this paper, a Maximum Entropy Markov Model (MEMM) for dialog state tracking is proposed to efficiently handle user goal evolvement in two steps. The system first predicts the occurrence of a user goal change based on linguistic features and dialog context for each dialog turn, and then the proposed model could utilize this user goal change information to infer the most probable dialog state sequence which underlies the evolvement of user goal during the dialog. It is believed that with the suggested various domain independent feature functions, the proposed model could better exploit not only the intra-dependencies within long ASR N-best lists but also the inter-dependencies of the observations across dialog turns, which leads to more efficient and accurate dialog state inference.

1 Introduction

The ability to converse with humans is usually considered the most important characteristic which defines the intelligent nature of a machine. In recent years, advanced approaches for handling different components within a spoken dialogue system have been proposed and studied. Both statistical inference methods for dialog state tracking and machine learning techniques (such as reinforcement learning) for automatic policy optimization are active domains of research, which implies that there are still many open challenges in this field that are worth being explored. One of such challenges is how to better exploit the ASR (Automatic Speech Recognition) N-best list when the top ASR hypothesis is incorrect. Furthermore, reasoning over different ASR N-best lists is also difficult since it is hard to decide when to detect commonality (when user repeats) and when to look for differences (when user changes her or his mind) among multiple ASR N-best lists. Another challenge is how to handle more complex user actions such as negotiating alternative choices or seeking out other potential solutions when interacting with the system.

This proposal presents a probabilistic framework for modeling the evolvement of user goal during the dialog (focusing on the shaded component *Dialog State Tracking* in Figure 1 that shows a typical diagram for a spoken dialog system), which aims to endow the system with the ability to model natural negotiation strategies, in the hope of leading to more accurate and efficient dialog state tracking performance.

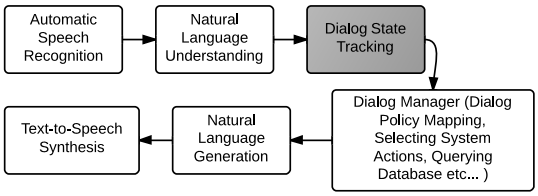

Figure 1: a typical spoken dialogue system

2 Unanswered Challenges for Spoken Dialog Systems

Due to the inevitable erroneous hypotheses made by the speech recognizer as well as the ubiquitous ambiguity existing in the natural language understand-

Proceedings of the NAACL HLT 2013 Student Research Workshop, pages 91–97,
Atlanta, Georgia, 13 June 2013. ©2013 Association for Computational Linguistics

ing process, it is impossible for a spoken dialog system to observe the true user goal directly. Therefore, methods to efficiently infer the true hidden dialog states from noisy observations over multiple dialog turns become crucial for building a robust spoken dialog system.

The POMDP (Partially Observable Markov Decision Process) framework has been proposed to maintain multiple dialog state hypotheses under uncertainty with automated dialog policy learning (Williams and Young, 2007; Henderson et al., 2008; Thomson and Young, 2010; Young et al., 2010). Although the original POMDP framework suffers difficulties of scaling up the model to handle real-world domains in practice, it provides a unified statistical framework for existing techniques with global optimization. Partition-based approaches (Gašić and Young, 2011; Williams, 2010; Young et al., 2010) attempt to group user goals into a number of partitions and won't split a partition unless when a distinction is required by observations. Due to this property, partition-based methods could have high scalability for more complex practical domains.

Bayesian network based approximate methods also emerged to tackle the complexity of representing and tracking multiple dialog states within probabilistic frameworks (Raux and Ma, 2011; Thomson and Young, 2010). In previous work, we presented a new probabilistic model – DPOT (Dynamic Probabilistic Ontology Trees) – to track dialog state in a spoken dialog system (Raux and Ma, 2011). DPOT captures both the user goal and the history of user dialog acts (user actions) using a unified Bayesian network. Efficient inference (a form of blocked Gibbs sampling) is performed to exploit the structure of the model. Evaluation on a corpus of dialogs from the CMU Let's Go system shows that DPOT significantly outperforms a deterministic baseline by exploiting long ASR N-best lists without loss of accuracy. At any point in the dialog, the joint distribution over the goal network represents the inferred dialog state about the user goal.[1] The goal network of DPOT does not expand per time slice for each turn but the evidence accumulates as the dialog progresses. Therefore the model becomes inefficient when users change their mind – user has to repeat multiple times in order to possibly trigger a goal change in the inferred dialog state.

System: How can I help you?
User: I would like a flight from Columbus to San Francisco.
System: I hear you say you want a flight from Columbus to San Francisco, is that correct?
User: Yes.
System: OK, at what time?
User: Saturday morning.
System: Leaving on Saturday morning. Searching available flights...
[some time later] I have found one flight from CMH to SFO leaving at 7am on Saturday morning. Do you want me to print out itinerary, make flight reservation or hear more information about the flight?
User: How about Sunday morning?
System: Sorry I am afraid I didn't catch that.
You can say print itinerary, book flight or more information.
User: None of them.
Do you have any flights leaving on Sunday morning instead?
System: Sorry, what can I do for you?
[system reset itself for a new dialog to recover from the failure]
User: ...

Figure 2: Example of user goal change: at the end of the dialog the user would like to explore alternative flights at a different time, but the dialog system did not expect such a user action, leading to a system failure

Current approaches often assume that user would have a fixed goal in his or her mind before conversing with the system and this single goal remains unchanged throughout the dialog. However, the key question we would like to raise here is that whether the assumption that a user would not change her or his mind during the dialog is reasonable or not in the first place.[2] Figure 2 shows an example where user goal evolves as the dialog moves on. In this example, the system did not catch the partial change of user goal and failed to return alternative answers given a new request from the user – now the fixed goal assumption has been challenged. Moreover, sometimes people do not even have a clear goal in their minds before they start speaking to the system (e.g., a user might want a flight from Columbus to San Francisco during the coming weekend, but the exact departure date depends on user's schedule as well as the price of the ticket.). From the example dialog shown in Figure 2, clearly it can be noticed that there are some useful hints or linguistic patterns – such as *How about ...?* and ... *instead?* – which could be extracted from the user's spoken language

[1] In the Let's Go bus information system, a user goal is decomposed into three concepts: `Bus` (the bus number), `Orig` (the origin stop) and `Dest` (the destination stop).

[2] It is true that for some simple domains such as luggage retrieval or call routing, users are less likely to change their mind.

as predictors for potential user goal change. We can then further use this predicted information (user goal changed or not) to better infer the true user goal and prevent a system failure or start over. In fact, it is this intuition that forms the basis of the proposed methods.

However, existing methods heavily rely on the assumption that user won't change her or his mind throughout the dialog. In order to keep the computations tractable in practice, POMDP-based methods often assume that user goal does not change during the dialog (Young et al., 2010). Moreover, within the POMDP framework there is a user action model which would suppress the weights of conflict observations for those slots which have already been filled – the intuition is that if a value for a certain slot has already been provided or observed, it is less likely that a new value will be provided again (based on the assumption of fixed user goal) and it is more likely to be a speech recognition error instead (Williams and Young, 2007). Furthermore, one of the claimed benefit for existing statistical dialog state inference methods is the ability to exploit the information lower down from ASR N-best lists by aggregating weak information across multiple dialog turns – the intuition is that overlapped consistent weak evidence is sometimes a useful hint for predicting the underlying true user goal (as illustrated in Figure 3) – again it implies that the user would repeatedly refine the same goal until the machine gets it.

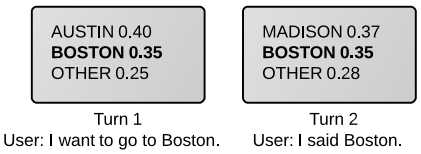

Figure 3: Given the fact that user action BOSTON has been repeatedly observed as `DEPARTURE_CITY` across the first two turns – although not at the top position of the ASR N-best list – existing statistical dialog state tracking algorithms would capture this pattern and put a strong bias on BOSTON as the inferred user goal.

It is true that putting such a constraint – assuming a fixed user goal during the dialog – simplifies the computational complexity, it also sacrifices the flexibility and usability of a spoken dialog system. Although one could think of some hand-crafted and ad-hoc rules such as explicit or implicit confirmation/disconfirmation to deal with sudden user goal changes during a dialog, it increases the number of dialog turns and makes the dialog system less natural and user friendly.

3 Spoken Dialog State Tracking with Explicit Model of User Goal Change

3.1 *BuildByVoice* Domain

In fact, there are many situations where frequent user goal changes would be highly expected (i.e. the user might try to *negotiate* with the system). These domains might include but not limited to finding nearby restaurants or hotels, searching for movies to watch, ordering food or online shopping, etc., in which users are very likely to explore different alternatives and their goals would probably change frequently as the dialog progresses.

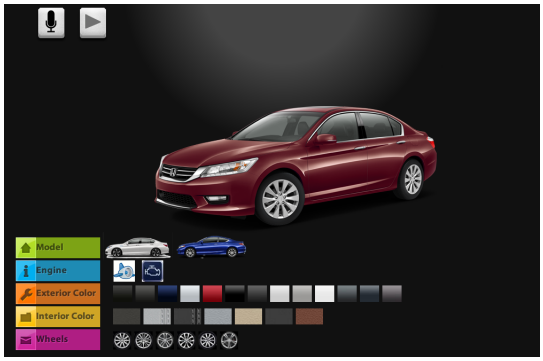

Figure 4: An experimental web interface prototype for *BuildByVoice* – a spoken dialog system aimed to assist potential car buyers to customize a car by voice

Considering one typical example among those domains – a spoken interactive system which could allow a user to configure a new car by speech (a prototype web interface of the *BuildByVoice* system is shown in Figure 4[3]) – one could imagine the user would tend to experiment many possible combinations of different configurations for a car. Indeed that is the purpose of having such a system so that users could preview the resulting effect before a real car is made. A *BuildByVoice* domain may consist of

[3] A baseline *BuildByVoice* system by using DPOT for dialog state tracking (without user goal change detection) is under implementation. The baseline system will be deployed to Amazon Mechanical Turk for initial data collection.

the following five independent concepts with their possible values listed as follows:[4]

Model: `Accord Coupe, Accord Sedan, Accord Plug-In, Civic Coupe, Civic Sedan, ...`[5]

Engine: `V4, V4 Turbo, V4 Sport, V6, V6 Turbo, V6 Sport, ...`

Exterior Color: `Toffee Brown, Coffee Brown, Candy Brown, Night Blue, Moonlight Blue, Midnight Blue, ...`

Interior Color: `Black Leather, Black Vinyl, Gray Leather, Gray Vinyl, Brown Leather, Brown Vinyl, ...`

Wheels: `17 inches Steel, 17 inches Alloy, 18 inches Steel, 18 inches Alloy, 18 inches Polished Alloy, ...`

In (Ammicht et al., 2007), the semantic representation of a spoken dialog system is augmented with a dynamic parameter that determines the evolution of a concept-value pair over time, which could be considered as early attempts for coping with user goal changes. However, the determined dynamic confidence score is used to make a hard choice for the candidate semantic values, i.e., determining the birth and death of the observed concept-value pairs. Thomson and Young (2010) introduced a new POMDP-based framework for building spoken dialog systems by using Bayesian updates of dialog state (BUDS). It accommodates for user goal changes by using a dynamic Bayesian network, but BUDS is generative rather than a discriminative model. Therefore it lacks the flexibility of incorporating all kinds of overlapping features – one of the advantages discriminative models have. Furthermore, BUDS assumes limited changes in the user goal in order to gain further efficiency. More recently, Gašić and Young (2011) introduces the explicit representation of complements in partitions which enables negotiation-type dialogs when user goal evolves during the dialog. However, the explicit representation of complements is used to provide existential and universal quantifiers in the system's response.[6] Also a special pruning technique is needed in their approach to ensure the number of partitions doesn't grow exponentially.

Therefore, new approaches for recognizing the event of user goal change and utilizing the goal change information to better infer dialog states have been proposed in the following two subsections 3.2 and 3.3.

3.2 Dialog State Tracking with Detected User Goal Change

Dialog state tracking is usually considered as the core component of a spoken dialog system where dialog manager uses the inferred dialog states to generate system responses (normally through a learned or hand-crafted policy mapping from dialog states to system actions). A specialized version of Maximum Entropy Markov Model with user goal change variable is proposed for dialog state tracking.[7] The most probable dialog state sequence as well as the most likely dialog state value for the latest turn can be inferred given the model. Figure 5 illustrates how the proposed model could infer dialog states of a single concept **Exterior Color** for a dialog of four user turns where the user changes her or his mind at the third dialog turn.[8]

For traditional dialog state tracking methods without user goal change model, the system would be quite confused by completely conflicting observed user actions starting from the third dialog turn. However, the proposed MEMM with user goal change detection could notice that the user has already changed her or his mind. Therefore the proposed model would not only trust more on the observed user actions for the current dialog turn, but also favor those transitions which lead to a different state value by increasing corresponding transition probabilities.

[4] More concepts could also be included such as **Accessories** or **MPG Level**, but only these five concepts are picked for demonstration purpose.

[5] Here *Honda* car models are used as an example.

[6] E.g., "Charlie Chan is the **only** Chinese restaurant in the center." or "**All** Chinese restaurants are in the center."

[7] Methods for detecting user goal change are described in Section 3.3.

[8] We assume every concept in the domain is mutually independent with each other and we model the user goal change separately for each concept.

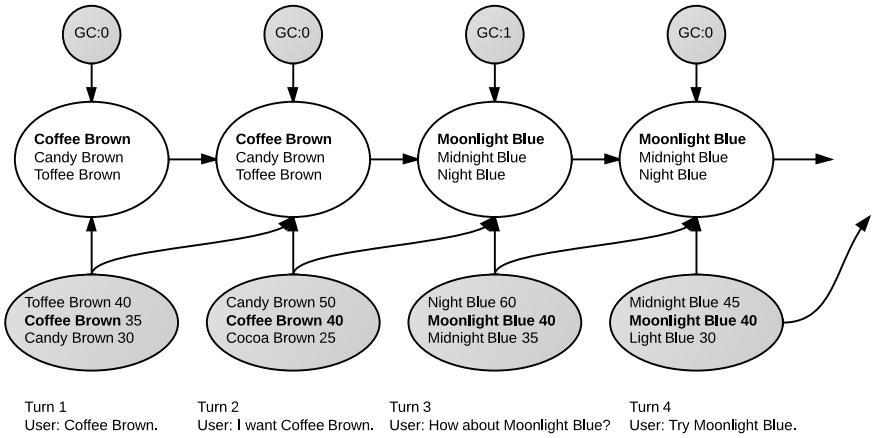

Figure 5: MEMM for dialog state tracking with explicit user goal change variable. A single concept **Exterior Color** from *BuildByVoice* domain is tracked by the model. The shaded nodes are observed user actions and the white nodes are hidden dialog states. The bold text in the observed nodes indicates the true user actions whereas the bold text in the hidden states shows the true dialog state sequence (in this case it is also the most probable decoded dialog state path inferred by the model).

A more formal description of the proposed MEMM is given as follows. The observations o_t (shaded nodes) consist of N-best lists of semantic speech hypotheses (or dialog acts) with confidence scores (scale from 0 to 100) for the current dialog turn hyp_t and previous turn hyp_{t-1} as well as the binary goal change variable gc_t for the current turn – essentially a context window of speech hypotheses including history:

$$o_t \quad = \quad \{hyp_{t-1}, hyp_t, gc_t\}$$

Typically the semantic speech hypotheses hyp_t are extracted concept-value pairs out of ASR results by using a semantic tagger (such as an FST (Finite State Transducer) parser or a segment-based semi-Markov CRF semantic labeler (Liu et al., 2012)). The hidden dialog state q_t (white nodes) represents the user goal for dialog turn t (such as a particular color `Moonlight Blue` for **Exterior Color** at time t). The individual probability of a transition from a state q_{t-1} to a state q_t producing an observation o_t is in a form of the following:

$$P(q_t|q_{t-1}, o_t) \quad = \quad \frac{\exp(\sum_{k=1}^{n} w_k f_k(q_{t-1}, q_t, o_t))}{Z(o_t, q_{t-1})}$$

Given labeled sequences of true dialog states (true user goal) for each turn, the corresponding observations and designed feature functions, we want to learn a set of weights w_k to optimize the discrimination among competing state values given the training data. In other words, the learning procedure involves searching in parameter space to maximize the following conditional likelihood:

$$P(Q|O) = \sum_{i=1}^{N} \prod_{t=1}^{T} \frac{\exp(\sum_{k=1}^{n} w_k f_k(q_{i,t-1}, q_{it}, o_{it}))}{Z(o_{it}, q_{i,t-1})}$$

where N is the number of training dialogs. MEMM can be trained with methods from the field of convex optimization and Viterbi decoding algorithm could be applied to MEMMs for inference (McCallum et al., 2000).

The proposed feature functions are as follows. The first feature function (1a) implies that if the user goal is not changed, the system should look for the common evidence across dialog turns.

$$f(q_t = v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}0 \ \& \\ & v{\in}\textit{common}(hyp_{t-1}, hyp_t) \\ 0 & \text{otherwise} \end{cases} \tag{1a}$$

where $\textit{common}(hyp_{t-1}, hyp_t)$ will return the overlapped values from the two N-best lists of dialog acts hyp_{t-1} and hyp_t. The second and third feature functions ((1b) and (1c)) are basically saying that if a user goal change has been detected, then we should expect a different state value, otherwise we should

remain the same value from previous dialog turn.

$$f(q_{t-1}=u, q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}0 \ \&\ u{=}v \\ 0 & \text{otherwise} \end{cases} \quad (1b)$$

$$f(q_{t-1}=u, q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}1 \ \&\ u{\neq}v \\ 0 & \text{otherwise} \end{cases} \quad (1c)$$

The intuition behind the following four feature functions (feature function (1d) to (1g)) is that if the user changes her or his mind then the model should trust more on the current observed user actions than those from previous turn; but if the user does not change her or his mind, we could then consider the observations from the past.

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}0 \ \&\ v{\in}hyp_{t-1} \\ 0 & \text{otherwise} \end{cases} \quad (1d)$$

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}1 \ \&\ v{\in}hyp_{t-1} \\ 0 & \text{otherwise} \end{cases} \quad (1e)$$

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}0 \ \&\ v{\in}hyp_t \\ 0 & \text{otherwise} \end{cases} \quad (1f)$$

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}1 \ \&\ v{\in}hyp_t \\ 0 & \text{otherwise} \end{cases} \quad (1g)$$

The last two feature functions ((1h) and (1i)) try to incorporate information from confidence scores – the higher the confidence score is, the more likely the hypothesis is to be correct.

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } v{\in}hyp_t \ \& \\ & \textit{confidence}_{hyp_t}(v){>}C \\ 0 & \text{otherwise} \end{cases} \quad (1h)$$

$$f(q_t=v, o_t) = \begin{cases} 1 & \text{if } gc_t{=}0 \ \& \\ & v{\in}hyp_{t-1} \ \& \\ & \textit{confidence}_{hyp_{t-1}}(v){>}C \\ 0 & \text{otherwise} \end{cases} \quad (1i)$$

where $\textit{confidence}_{hyp_t}(v)$ returns the confidence score for value v in the speech hypotheses N-best list hyp_t and C is an empirical constant threshold range between 0 to 100 obtained from the training corpus.

3.3 User Goal Change Detection with Linguistic Features and Dialog Context

In previous subsection 3.2, we assume we already know whether or not user changes her or his mind at each dialog turn, whereas this subsection we discuss the possible approaches on how to detect a user goal change. Detecting user goal changes during a dialog could be cast as a binary classification problem where class 0 means no goal change and class 1 indicates user changes her or his mind during a dialog turn. Candidate machine learning algorithms including MLP (Multi-layer Perceptron), SVM (Support Vector Machine) or Logistic Regression could be applied to this binary classification problem in a supervised manner. The input features might be extracted from user utterance transcription[9] and the corresponding ASR N-best list for each dialog turn. As mentioned in Section 2, the language patterns found in the user utterances as presented in the example dialog (shown in Figure 2) forms the intuition for linguistic features to identify user goal change. The dialog context such as last system action could also be included as useful hint for predicting a potential user goal change – user is likely to change her or his goal if system returns empty results for a request. Also other helpful features could include bag of words model, n-grams, prosodic features (e.g., a pitch change or initial pause) and parsed features (e.g., WH questions). Baseline system such as key word spotting based approach (i.e. look for *How/What about* in a sentence) could also be implemented for performance comparison.[10]

4 Conclusion

By modeling the user goal change in a probabilistic framework, the proposed approach should better exploit the mutual information buried deep in the ASR N-best lists across dialog turns, which leads to more robust and accurate dialog state estimation. With the ability to predict and handle user goal change, proposed techniques provide a bottom-up solution for managing negotiation style dialogs and not only should produce more efficient and natural conversations but also open up new possibilities for automated negotiation dialog policy learning.

[9] At test time, this could be approximated by the top hypothesis in the ASR N-best list.

[10] A detailed list of proposed features is omitted due to space limit.

References

Egbert Ammicht, Eric Fosler-Lussier, and Alexandros Potamianos. 2007. Information seeking spoken dialogue systems—part i: Semantics and pragmatics. *Multimedia, IEEE Transactions on*, 9(3):532–549.

M. Gašić and S. Young. 2011. Effective handling of dialogue state in the hidden information state pomdp-based dialogue manager. *ACM Transactions on Speech and Language Processing (TSLP)*, 7(3):4.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, 34(4):487–511.

J. Liu, S. Cyphers, P. Pasupat, I. McGraw, and J. Glass. 2012. A conversational movie search system based on conditional random fields. In *INTERSPEECH*.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, volume 951, pages 591–598.

A. Raux and Y. Ma. 2011. Efficient probabilistic tracking of user goal and dialog history for spoken dialog systems. In *Twelfth Annual Conference of the International Speech Communication Association*.

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

J.D. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

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

S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.