A Discriminative Classification-Based Approach to Information State Updates for a Multi-Domain Dialog System

Dilek Hakkani-Tür, Gokhan Tur, Larry Heck, Ashley Fidler, Asli Celikyilmaz

Microsoft, Mountain View, CA

{dilek,gokhan.tur,asli}@ieee.org,{larry.heck,ashfid}@microsoft.com

Abstract

We propose a discriminative classification approach for updating the current information state of a multi-domain dialog system based on user responses. Our method uses a set of lexical and domain independent features to compare the spoken language understanding (SLU) output for the current user turn with the previous information state. We then update the information state accordingly, employing a discriminative machine learning approach. Using a data set collected from our conversational interaction system, we investigate the impact of features based on context dependent and context independent SLU tagging schemas. We show that the proposed approach outperforms two non-trivial baselines, one based on manually crafted rules and the other on classification with lexical features alone. Furthermore, such an approach allows the addition of new domains to the dialog manager in a seamless way.

Index Terms: multi-domain spoken dialog systems, multi-turn spoken language understanding, learning information state updates

1. Introduction

Goal-oriented conversational interaction systems aim to automatically identify the intention of the user as expressed in natural language, extract associated arguments or slots, and take actions accordingly to satisfy the user's requests. In such systems, if spoken, the speaker's utterance is typically recognized using an automatic speech recognizer (ASR). Then the intent of the speaker along with its arguments are identified from the user's utterance using a spoken language understanding (SLU) component. Finally, a dialog manager (DM) interacts with the user (not necessarily in natural language alone) and a back-end of structured or unstructured information sources help the user achieve the task that the system is designed to support.

More formally, at each turn, a user's input utterance, X_i, is converted into a task-specific semantic representation of the user's intention, I_i. This spoken language understanding step mainly involves semantic parsing and interpretation; readers may refer to [1] for an exhaustive survey on this topic.

The dialog manager then decides on the most appropriate system action, A_i. In statistical approaches, this decision is made based on the expected reward over belief [3, 4] or information states, which are estimated using I_i and the semantic context, C_u, that includes the previous information state of the system, S_{i-1}, user specific meta-information, such as geolocation and personal preferences, and other contextual information. For example, if the user clicks on a map on the screen and says "*How much is the cheapest flight from here to New York?*," the system should be able to interpret the intent and the associated arguments, such as:

Domain=*flight*, Intent=*get_ticket_price*
Cost_Relative=*cheapest*,
Origin=(latitude,longitude),
Destination=*New York*

More formally, a statistical model may estimate S_i as:

$$\hat{S}_i = \underset{S_i}{\operatorname{argmax}}\, P(S_i|I_i, C_u)$$

The statistical modeling of information or belief state updates is relevant for various tasks. One of the earliest statistical approaches to multi-turn interpretation is the statistical discourse model by Miller *et al.* [2] that introduced a mapping from the pre-discourse meaning and the previous meaning (as determined after the previous user turn) to the post-discourse meaning. A set of five operations that are used to create the post-discourse meaning is defined, and decision trees are trained to determine the operation. These operations are defined for each element of the semantic meaning representation, and hence a separate decision tree is trained for each of these elements. Later on, Levin and Pieraccini [3] proposed using Markov Decision Process (MDP) as a dialog model. MDPs assume that dialog states are observable; hence, they do not account for any uncertainty in the dialog history or the user state. The application of partially observable Markov decision processes (POMDP) were suggested for dialog modeling [4] to handle uncertainty.

Our work is in line with that of Traum and Larsson [5], who introduced the information state update approach. The information state of a system is defined as the information necessary to distinguish a dialog from others, representing the cumulative additions from previous actions in the dialog. The key to this approach is the *update* of information states as the dialog progresses. In this work, we focus on a set of updates that integrate the SLU output from the current turn with the previous information state. Our approach is also similar to the work of Miller *et al.*; however, our update operations work on, but are not limited to, the full SLU representation level, allowing the training of a single, generic classifier. We also allow for several different ways of merging slot values, such as with disjunction or conjunction [6].

One major contribution of this study (in comparison to previous statistical models of dialog) is the focus on multi-domain conversational interaction systems. In a multi-domain dialog system, some conversational turns can be interpreted as belonging to multiple possible domains. For example, depending on the context, the user utterance "*How about tomorrow?*" might refer to either *flight departure date* or *date for weather inquiry*. Each user turn can be treated as an attempt to either switch domains and/or intents or to add constraints to the ones specified

User: Show me flights for San Francisco to Seattle tomorrow
System: Flights from SFO to SEA tomorrow: 10:00am AS 203 11:00am SW 302 ...
User: Which ones are in the afternoon?
System: Flights from SFO to SEA tomorrow, in the afternoon: 1:24pm AS 249 ...
User: How about from San Jose California?
System: Flights from SJC to SEA tomorrow, in the afternoon: 2:32pm UA 342 ...

Table 1: Example spoken interaction with the dialog system.

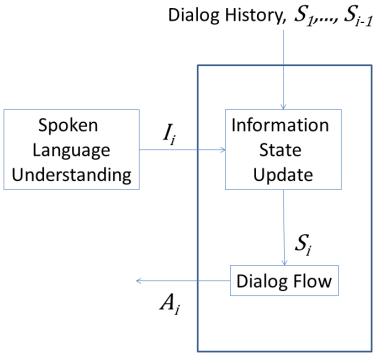

Figure 1: A conceptual architecture of the conversational interaction system, focusing on the dialog manager component.

by the previous utterances. In this paper, we propose a machine learning approach for determining how to update the information state based on these user queries. Such an approach allows for the use of a discriminative classifier and integrates domain and task independent features. Hence, in contrast to other learning approaches to DM, when a new domain is introduced to the system in our approach, one needs only to train the spoken language understanding models, and the estimation of update rules would work seamlessly.

Furthermore, compared to previous DM models, our approach enables flexibility in DM modeling, resulting in more robust DM system for multi-domain systems. One important issue that arises in conventional DMs under multi-domain settings is that, the models are trained under the assumption that the domain information is provided a priori or committed in SLU. Our proposed DM state update approach is novel in that we not only determine the type of the update rules but also interpret the domain of the utterance (especially for vague utterances) in the current turn.

In Section 2, we present related work, focusing on the information state update approach to dialog management [5]. Then we describe our multi-domain spoken dialog system and the classification-based approach to determining information state updates. In our experiments, we show that lexical and domain independent SLU-based features can be used with discriminative classification to determine which information state update to apply.

2. Brief Overview of The Dialog System

We focus on multi-domain dialog systems, where users can interact with the system mainly using speech. Table 1 shows a sample interaction with the system, whether user is seeking information about flights.

The conceptual architecture of the conversational interaction system used in this study is depicted in Figure 1. The goal of SLU in our system is to extract the domain, intent and a set of slots from user utterances [1]. For example, for the first user utterance in the example dialog, *"Show me flights for San Francisco to Seattle tomorrow"*, the domain is *flights*, the user intent is *find flights*, and the slots are *origination city*, *destination city*, and *departure date*, with the slot values *San Francisco*, *Seattle*, and *tomorrow*, respectively. More details about the specific domain/intent detection and slot filling approaches can be found in [7] and [8]. Once the domain, intent and slots have been determined, the dialog manager either initiates a new information state, or integrates the user utterance into the previous state, launches an information source look-up when applicable, and decides on the next system action. Our current dialog system uses a manually-built, finite-state-based dialog flow for determining the next system action. Next, a template-based natural language generation (NLG) component is used to request more information from the user or to provide feedback to the user about the results being displayed and about system's interpretation of the user's utterance. The NLG output is presented to the user both visually, as text on the screen, and through text-to-speech synthesis.

Other multi-modal information processing tasks (such as gesture, click, or touch) can be built on top of this basic framework. In this paper, we assume the input to the dialog manager is the interpreted and normalized multi-modal SLU output and focus on the statistical modeling of the information state update, instead of these lower-level yet non-trivial tasks.

3. Multi-Turn Interpretation and Information State Update

Our approach assumes an information state representation that includes the same components as the SLU representation: domain, intent and slots, and views information state updates as the estimation of the current information state, given the previous information state and the SLU output for the current turn.

3.1. State Update Actions and Classification

We define a set of 12 state update actions that aim to either integrate the information in the current user turn with the previous information state (*Add, Override-Domain, Override-Intent, Override-Slot, Keep, Repeat, Add-AND, Add-OR,* and *Add-ButNot*), to initiate a new information state (*Initiate*), or to *go back* to a previous information state in the dialog history (*Pop* and *Reset*). The dialog history is represented as a last-in, first-out stack, and the decision of whether to push a new information state onto the stack, or to pop the most recent information state off is determined by the state update actions. Table 2 presents examples of some of these state update actions, with a sample user-system interaction involving two user turns, with four alternative user responses as the second user turn. In the first alternative, the user adds a new slot value for *non-stop* flights, in the second one the user aims to override the value of the date slot, and so on.

User turn	Update Action	Information State
U1: Flights from SFO to SEA tomorrow	Initiate	Domain = flights, Intent = find flights Origin = SFO, Destination = SEA, Date = 3/23/2012
U2.1: Which ones are non-stop?	Add	Domain = flights, Intent = find flights Origin = SFO, Destination = SEA, Date = 3/23/2012, NumStops=non-stop
U2.2: How about on Saturday?	Override-slot	Domain = flights, Intent = find flights Origin = SFO, Destination = SEA, Date = 3/24/2012
U2.3: From San Francisco to Seattle	Repeat	Domain = flights, Intent = find flights Origin = SFO, Destination = SEA, Date = 3/23/2012
U2.4: Show also the ones one Saturday	Add-OR	Domain = flights, Intent = find flights Origin = SFO, Destination = SEA, Date = 3/23/2012 OR 3/24/2012

Table 2: Examples of alternative second turns (U2.1, U2.2, ...) of the user, related state update actions, and resulting information states.

Users are allowed to undo their last turn, by simply saying "go back" or one of its variants, or by resetting the session. These are covered by the SLU as special commands and map to specific state update actions.

Once the type of update is determined, the new information state is built accordingly. For example, if the user adds a new slot value, the new state is formed by adding the slots and their values from the current turn to the previous information state.

In a multi-domain dialog system, some user utterances do not explicitly specify a single domain. For these ambiguous cases, we experiment with two SLU tagging schemas: one is context sensitive, i.e. domains are transferred from previous turns, and the other is context independent, meaning that vague utterances are explicitly marked as such. For example, in the context dependent SLU, the utterance "*How about tomorrow?*" is tagged as belonging to *flight* or *weather* domain if the dialog context implies that. In the context-independent SLU, such an utterance is tagged as *vague*, and then the DM decides on its domain using context. We study the effect of such high-level SLU decisions on the DM state estimation.

3.2. Classification Features

The current system uses two types of features, lexical features and features obtained by comparing the SLU output, I_i, for the current turn with the previous information state, S_{i-1}.

3.2.1. Lexical Features

Lexical features are expected to be good indicators of users adding or overriding previously introduced slot values. For example, sequences such as *how about*, *which one(s)*, *instead* usually refer to items that the user already specified in a previous turn. We extract all word n-grams from user utterances to learn such implications.

3.2.2. SLU-derived features

The SLU output S_i provides three kinds of complementary information, the estimated domain, intent, and slots. Instead of using them as is, we have extracted features comparing them with the previous information state, as described below.

- *Domain-related features*: aim to compare the domain values and scores of the top domains in I_i and S_{i-1}. We denote the most probable domain for the previous information state with ds_{i-1}, and the one for the current SLU output with di_i:

$$ds_{i-1} = \underset{j}{\operatorname{argmax}}\, score(d_j|S_{i-1})$$

$$di_i = \underset{j}{\operatorname{argmax}}\, score(d_j|I_i)$$

  where $score(d_j|I_i)$ and $score(d_j|S_{i-1})$ denote the scores assigned to the domain $d_j \in D$ by SLU and information state updates, respectively, where D denotes the set of possible domain categories.

- *Intent-related features*: aim to compare the intent values and scores of the top intents corresponding to the top domains in I_i and S_{i-1}, di_i and ds_{i-1}, in a similar way to the domain features.

- *Slot-related features*: aim to check whether the SLU output for the new turn, I_i, is introducing, overriding, or repeating a slot value when compared with the slots and their values in S_{i-1}. Similar to the intents, slots are estimated for each domain category, hence we have two sets of features that compare the slot values for the top domains of both I_i and S_{i-1}, di_i and ds_{i-1}.

Since, it is not clear what defines a session from the system's viewpoint, knowledge of session and sub-session beginnings and endings and turn ranks were not used when extracting features in our experiments. The feature set used can also be extended to include information presented to the user and the system's turn.

3.2.3. Statistical Modeling of Information State Update

Once the lexical and SLU-derived features are extracted, one may employ any classification approach. A central parameter is deciding on whether this is a sequence or sample classification task. In this study, we only performed experiments using a discriminative sample classification algorithm, namely Boosting, which is known to have superior performance for discrete valued features as used in this work.

4. Experiments

4.1. Data Sets

The data set used in our experiments consists of interactions with the multi-domain dialog system collected from around 30

Features	Error Rate (CD)	Error Rate (CI)
Baseline	34.3%	
1: Word n-grams	27.5%	
2: 1 + Domain Features	25.5%	22.8%
3: 2 + Intent Features	23.2%	22.3%
4: 3 + Slot Features	**22.1%**	**21.5%**

Table 3: State update action detection error rates with context dependent (CD) and context independent (CI) SLU with various incrementally-introduced sets of features.

users, in 274 sessions, (corresponding to 604 sub-sessions, defined as the times when the user shifted his/her goal without explicitly requesting that the system start a new session), and 2,638 user turns. The sessions range from 2 user turns to 60 user turns, with an average of 9.6 user turns per session. Every user turn is manually transcribed and labeled with the information state update actions the system is expected to take. The statistical SLU models are used to predict the domain, intent and slots of each turn. The training set of SLU models included 20,000 utterances from 25 domains, manually transcribed and annotated for SLU tasks. Users were not given any specific instructions or scenarios before or during the collection, but were familiar with the capabilities of the system and the covered domains.

4.2. Results

In this work, we use icsiboost [9], an implementation of the Adaboost algorithm [10], for classification when determining the type of the information state updates. We perform n-fold cross validation experiments, using a session as the test set in each fold. Table 3 presents the resulting classification error rates at the turn-level, when the word n-grams, domain, intent, and slot features are incrementally introduced with the context dependent and context independent SLU tagging approach. Word n-grams in these experiments include all unigrams, bigrams, and trigrams from the user utterances. The baseline experiment assumes that all system commands, such as explicit user requests to initiate a new search, are recognized correctly, and that the rest of the utterances are assigned the majority class, which, in this case, is "initiate a new information state".

As seen in these results, word n-grams are useful in determining the information state update, and the addition of each new type of feature leads to an improvement over the previous results. The inclusion of SLU related features results in over 20% relative error reduction over the baseline based on word n-gram features, from 27.5% to 21.5% with the context dependent SLU. The error rates with the context independent SLU tagging scheme (that tags utterances that do not have explicitly stated domains as vague), are lower than the results with the same set of features derived from the context dependent SLU, however the difference between these results decreases as we add more features and the dominance of domain features in the feature set is reduced.

The weak learners chosen by boosting include all types of features: terms or phrases that imply commands, or that imply the utterance is referring to a previously defined item, such as *which* and *what else* are amongst the lexical features; and chosen SLU features include thresholds for classification scores, and features that check if the SLU domain, intent, and slots are overriding or repeating the ones from the previous utterances.

5. Conclusions

We have presented a discriminative classification approach for updating the current information state of a multi-domain dialog system. The proposed method uses a set of lexical and domain independent features to compare the spoken language understanding output for the current user turn with the previous information state. We show that this approach outperforms two non-trivial baselines, one based on manually crafted rules and the other on classification with lexical features alone.

Furthermore, such an approach allows the addition of new domains in a seamless way to the dialog manager. Adding a new domain requires only training SLU domain/intent detection and slot filling models. While the approach can work in a standalone fashion, it can also be used in a complementary way to other approaches, such as reinforcement learning or POMDPs, by providing the information state update actions as features or prior information that can be integrated as constraints or for reducing the state space.

Our aim in these experiments was mainly to see if such an approach was viable and the feature sets can be extended to include n-best results from speech recognition and understanding as well as system prompts, as part of the future work. Moreover, the use of a sequence classifier may further improve the results. The proposed method can also be used in a complementary way to other statistical dialog management approaches. Our future work involves incorporating reinforcement learning into this framework, learning from the implicit feedback provided by the users.

Acknowledgments: Authors would like to thank Ruhi Sarikaya, Daniel Boies, Alexandre Rochette, Jason Williams and Dan Bohus for useful discussions.

6. References

[1] G. Tur and R. De Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, John Wiley and Sons, New York, NY, 2011.

[2] Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz, "A fully statistical approach to natural language interfaces," in *Proceedings of ACL*, 1996.

[3] Esther Levin and Roberto Pieraccini, "A stochastic model of computer-human interaction for learning dialogue strategies," in *Proceedings of Eurospeech*, 1997.

[4] Jason D. Williams and Steve J. Young, "Scaling up pomdps for dialog management: the "summary pomdp" method," in *Proceedings of ASRU Workshop*, 2005.

[5] David Traum and Staffan Larsson, "Current and new directions in discourse and dialogue," chapter The Information State Approach to Dialogue Management, pp. 325–353. Kluwer, 2003.

[6] Paul A. Crook and Oliver Lemon, "Lossless value directed compression of complex user goal states for statistical spoken dialogue systems," in *Proceedings of Interspeech*, 2011.

[7] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Exploiting web search query click logs for utterance domain detection in spoken language understanding," in *Proceedings of ICASSP*, 2011.

[8] Gokhan Tur, Dilek Hakkani-Tür, Dustin Hillard, and Asli Celikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling," in *Proceedings of Interspeech*, 2011.

[9] Benoit Favre, Dilek Hakkani-Tür, and Sebastien Cuendet, "Icsiboost," `http://code.google.come/p/icsiboost`, 2007.

[10] Robert E. Schapire and Yoram Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.