Dynamic Language Modeling using Bayesian Networks for Spoken Dialog Systems

Antoine Raux[1], *Neville Mehta*[2], *Deepak Ramachandran*[1], *Rakesh Gupta*[1]

[1]Honda Research Institute USA
[2]School of Electrical Engineering and Computer Science,
Oregon State University

araux@honda-ri.com, mehtane@eecs.oregonstate.edu,
dramachandran@honda-ri.com, rgupta@honda-ri.com

Abstract

We introduce a new framework employing statistical language models (SLMs) for spoken dialog systems that facilitates the dynamic update of word probabilities based on dialog history. In combination with traditional state-dependent SLMs, we use a Bayesian Network to capture dependencies between user goal concepts and compute accurate distributions over words that express these concepts. This allows the framework to exploit information provided by the user in previous turns to predict the value of the unobserved concepts. We evaluate this approach on a large corpus of publicly available dialogs from the CMU Let's Go bus information system, and show that our approach significantly improves concept understanding precision over purely state-dependent SLMs.

Index Terms: speech recognition, speech understanding, language modeling, spoken dialog systems

1. Introduction

As Automatic Speech Recognition (ASR) technologies mature, their field of application has widened from pure transcription tasks to complex interactive systems. In such systems, ASR does not operate in the void but rather within a rich context comprising the interaction history and external knowledge sources, and even the real world environment itself for embodied systems. Context is inherent in a user's goals and how they are expressed. For example, a person calling a restaurant information system on a Saturday night might be more likely to ask about bars than someone calling on a weekday at lunchtime. The ASR module of interactive systems should be able to leverage such contextual information to improve language modeling, and consequently the accuracy of speech understanding.

There are two important issues to tackle in order to predict user utterances based on context. First, one needs to model the influence of context on user goal. Second, the ASR's language model should be conditioned on the user's goal model. Focusing on information access dialog systems, we propose a new language modeling framework where the probabilities of relevant content words are estimated using a Bayesian Network (BN). The BN offers a principled way to incorporate information from previous user utterances as well as prior domain knowledge in order to estimate the user goal and compute contextual word probabilities. In Section 2, we review previous work on exploiting interaction context to improve speech recognition. Section 3 describes our proposed approach. The evaluation results on the CMU Let's Go corpus are given in Section 4 and discussed in Section 5.

2. Previous Work

The most common approach for exploiting context to improve ASR involves employing state-dependent language models, that is, using different statistical language models (SLMs) for different states of the dialog. Typically, such models are trained on a corpus of transcribed dialogs from the target system where user utterances are clustered according to the previous system prompt [1, 2], the dialog state [3], or the utterance topic [5]. Solsona et al. [6] present an alternative approach where the state-dependent language models are finite-state grammars rather than SLMs. A common aspect of these methods is that they aim at better predicting which concepts the user might provide in their next utterance, but not what the specific values of those concepts might be (i.e., what the user's specific goal is). This is problematic because understanding the user intention is key to the success of a dialog system. Gruenstein et al. [7] describe an approach where dialog context is used to anticipate both which concept the user might provide (through what the authors call weak contextual cues) as well as the values they might provide for these concepts (through strong contextual cues). However, strong cues are limited to cases where the system offers a limited list of choices to the user (e.g., "I've got a flight at two o'clock and another at four thirty."). Although useful, this approach does not allow the inclusion of other sources of information besides the previous system prompt that might help predict what the user will say. Finally, in the context of a train timetable system, Wiggers and Rothkrantz [8] propose to use class-based SLMs where the intra-class distribution of the destination station is conditioned on the departure station. The probability of two stations appearing in a request is derived from a large corpus of telephone timetable queries to human operators. This prior information significantly improves ASR accuracy. However, this approach is very specific to a case where two concepts are strongly related. Our approach, described in the next section, can be seen as generalizing this work to any number of concepts and relationships among them.

3. Dynamic Language Modeling

This section motivates the assumptions behind our dynamic language model and describes the application of BNs.

3.1. Model Structure

For explanatory purposes, we use the dialog in Figure 1 where the dialog system is trying to infer the probability of the word "downtown" in turn 4.

```
System : What can I do for you?       (1)
User   : I need the 28X schedule.     (2)
System : Where are you leaving from? (3)
User   : Leaving from downtown.       (4)
```

Figure 1: *Example dialog in the Let's Go domain.*

Assuming that the dialog system's aim is to fulfill the user's information-gathering goal, there are three broad contextual factors that influence the occurrence of the i-th word W_i in an n-word user utterance.

1. Linguistic variability L_i captures the dependencies between W_i and the previous words in a given utterance due to the syntactic patterns of natural language. In the example, it is more likely that "Leaving from" is followed by "downtown" than "are". This type of context is captured by standard n-gram models and finite-state grammars.
2. Discourse structure D captures the dependencies between turns of a dialog. In the example, the fact that the system asks the user about their departure place in turn 3 increases the probability that the user will provide that information in turn 4. This type of context is readily captured by the state-dependent SLMs described in Section 2.
3. User goal $G \in G_1 \times G_2 \times \cdots \times G_m$ where G_i represents an atom of information (such as "DeparturePlace") with its domain of values ("Carnegie Mellon", "airport" etc.). G intrinsically determines the user's choice of words. In the example, if the system is aware that the user's goal is to obtain the time of the next 28X bus from downtown to the airport, then it knows that the probability of "downtown" appearing in turn 4 is significantly higher than that of "Carnegie Mellon" or "airport".

Standard state-dependent language models capture linguistic and discourse variability but do not leverage information from past turns to predict the user goal and the probability of concept-values like "DeparturePlace=Downtown". In contrast, our language model, shown graphically in Figure 2, explicitly captures all three contextual factors, and applies a class-based n-gram model [9], where word class C_i isolates L_i and D from W_i and G. We assume a one-to-one mapping between the classes in the SLM and the goal concepts $\{G_j\}$, and let the goal context dynamically adjust the word occurrence probabilities. While the system does not know G at the beginning of the dialog in practice, every user utterance does provide some evidence about its component concepts. From the model,

$$P(W_i|L_i, D, G) = P(C_i|L_i, D) \cdot P(W_i|C_i, G). \quad (1)$$

Here, linguistic variability L_i is captured by a standard class-level n-gram, and the local discourse context D is modeled using standard context-dependent SLMs. Since there is a one-to-one mapping between word classes and goal concepts, if we assume that the class C_i of W_i corresponds to concept G_j, then $P(W_i|C_i = G_j, G) = P(W_i|C_i = G_j, G_j = g)$, that is, it only depends on the value g of G_j in G. Moreover, as the value g for concept G_j can be expressed in several ways (for instance, given the goal "DeparturePlace=Airport", the departure place can be expressed as "the airport" or "the international airport"), this probability covers all such words or phrases. Most of the time, g will not be known until the user utters it. Therefore, instead of assuming a specific value for G_j, we marginalize it out:

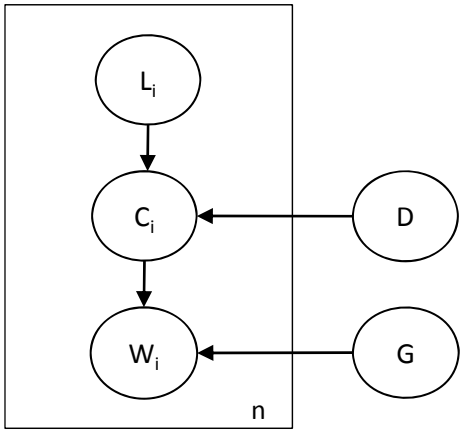

Figure 2: *Graphical representation of the proposed language model.*

$$P(W_i|C_i = G_j) = \sum_{g \in G_j} P(G_j = g) \cdot P(W_i|C_i = G_j, G_j = g) \quad (2)$$

The aim of the user goal model is to obtain an informative value for $P(G_j)$. As has already been done in belief tracking [13], we employ a Bayesian Network (BN) to model the user goal. G is represented as a set of interconnected random variables, each variable representing a concept or an external source of information (e.g. the time of the day when the dialog occurs). During the dialog, each user utterance is treated as evidence over one or more of the concepts. By leveraging prior knowledge on the relationship between concepts, we can compute accurate marginal distributions for all nodes of the BN (including yet-to-be-observed nodes). These marginal distributions can in turn be plugged into equation 2 as the value for $P(G_j)$.

3.2. Runtime Algorithm

Initially, the marginal distributions of each concept correspond to overall prior knowledge. After each user utterance, the system extracts the list of concept-value pairs from the top ASR hypothesis. For each concept-value pair (G_j, g_j), the system attaches a boolean observation node O_j to the matching concept node G_j. The observation node's value is set (observed) to $true$. The conditional probability distribution (CPD) of O_j given G_j is:

$$\begin{aligned} P(O_j = true|G_j = u) &= \epsilon, \forall u \neq g_j \\ P(O_j = true|G_j = g_j) &= 1 - \epsilon \end{aligned}$$

with $\epsilon \ll 1$. This approach is equivalent to directly observing the value of G_j except that it allows for conflicting evidence to be put in the network (e.g. both a neighborhood and a place that is not in the neighborhood could appear in the ASR results of a dialog). An alternative approach is to only add an observation to the BN once the corresponding concept-value pair has been explicitly confirmed by the user. This latter method is more conservative in that it requires more dialog turns because every

concept needs to be confirmed, but it avoids adding erroneous evidence to the BN.

Once the evidence has been added to the BN, standard inference algorithms, either exact or approximate, can be used to compute the resulting marginal probabilities of the concept nodes. We then update the intra-class probabilities of the SLM accordingly, before the next user utterance is recognized.

4. Evaluation

4.1. The Let's Go Corpus

We evaluated our approach on data collected with the CMU Let's Go telephone-based bus information system [10]. CMU Let's Go provides bus schedules for the Pittsburgh metropolitan area to the general population. All dialogs start with a general prompt ("What can I do for you?") after which the user can provide the whole ("When is the next bus from downtown to the airport?") or only part of their request ("The schedule for the 28X"). In subsequent turns, the system asks explicitly for the missing pieces of information and provides the relevant results when possible. Internally, the system can understand 6 concepts from the user: Bus Route (optional), Departure Neighborhood, Departure Stop, Arrival Neighborhood, Arrival Stop, and Travel Time. If the user specifies a neighborhood instead of a specific stop, the system uses a default stop for that neighborhood. In order to limit the impact of ASR errors, whenever the system understands a concept from the user, it asks for an explicit confirmation from the user ("From the airport. Did I get that right?").

The CMU Let's Go Corpus features 4305 dialogs, which represents a transcribed subset of all the dialogs recorded between March 2005 and October 2008. For the purpose of this research, we divided the corpus in a training set of 4010 dialogs (all the dialogs from 03/05 to 05/08, for a total of 48,237 non-empty utterances), a development set of 52 dialogs (from 09/24-09/25/08, 790 utterances), and a test set of 208 dialogs (from 09/26-09/30/08, 1931 utterances). We parsed the transcripts of each turn using the system's grammar and parser[1] and labeled each turn with the concept-value pairs it contains. For example, "DepartureNeighborhood=Downtown" is the concept-value tag for the utterance "I'm leaving from downtown". To deal with place name variants, we normalized all stop and neighborhood names by removing common words and ordering the remaining words lexicographically. For example, "FORBES_MURRAY" is the normalized form of both "Murray and Forbes" and "Forbes at Murray Avenue". Based on the turn-level semantic labels, we derived dialog-level user goals.

4.2. System Overview

Based on our knowledge of the domain, we designed a simple Bayesian Network to capture dependencies among place concepts in the Let's Go domain. Figure 3 shows the structure of the network. It is a tree-structured network that captures the probability that a neighborhood contains the departure/destination stop of a trip on a given bus route as well as the probability that the user looks for a specific stop in given a neighborhood.

All the parameters of the model described in Section 2 were estimated using the training set. First, we trained context-dependent class-based trigrams with modified Knesser-Ney smoothing using the SRI LM toolkit [12]. Second, we estimated

[1]Let's Go uses the Phoenix robust parser [11].

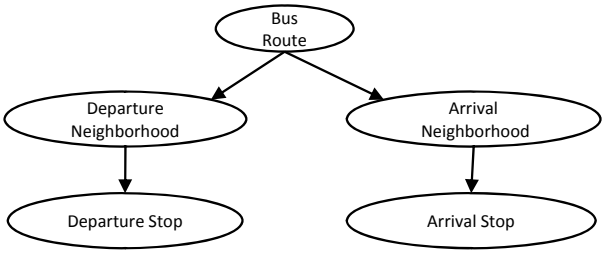

Figure 3: *User Goal Bayesian Network for Let's Go.*

the goal-dependent class unigram distributions $P(W_i|C_i, G)$ for all different concept values, using maximum likelihood. Finally, we counted the values for each concept and computed the conditional probability distributions (CPDs) of the BN based on their frequencies.

At runtime, speech recognition is performed using the Sphinx 4 recognition engine [14] using the original Let's Go acoustic models. Bayesian inference within the goal model is performed using belief propagation [15] as implemented in the Probabilistic Network Library [16].

We implemented two different versions of the system: **NoConfirm** uses all concept-value pairs from each turn's top ASR hypothesis to update the goal model; **Confirm** simulates a system that performs explicit confirmation on the concepts. In this case, the ASR hypothesis is filtered by removing those concept-value pairs which do not appear in the ground truth (similarly to what happens when a user responds negatively to an explicit confirmation). All systems use the Let's Go's robust parser and hand-written grammar to extract the concept-value pairs from the ASR output.

4.3. Baseline and Oracle Conditions

We compare the performance of our algorithm to two systems. First, the baseline system uses a standard state-dependent class-based SLM. The SLM's intra-class word probability distributions are trained on the training set but are kept static during the interaction. This is a good proxy for standard practice in SLM-based spoken dialog systems today.

To estimate the impact of misrecognitions of previous concepts on the performance of the model, we introduced an oracle system. This version uses the same goal model as the proposed approach but, after each turn, the belief is updated using all the ground truth concept-value pairs, rather than the ASR results.

4.4. Results

Table 1 shows the precision and recall for the place concepts in the Let's Go domain. The word error rate (WER) being almost identical across all conditions is not surprising because the proposed approach focuses on improving the accuracy of the concept values which are normally a small portion of the words in the utterances. These WERs are in accordance with previous results published for spoken dialog systems. In particular, they match the results previously reported for the Let's Go system [10].

For concept understanding, the baseline performance is fairly low, which reflects the challenging nature of this real-world task, due to the wide range of line qualities (landline vs wireless), environments (indoor vs outdoor) and speaker characteristics (age, gender, accent, experience with ASR).

The oracle system improves precision by 6.5% absolute while recall is not statistically significantly different from the baseline[2]. The **NoConfirm** condition yields 4.0% improvement in precision but a statistically significant loss (3.5%) in recall. This indicates that, as-is, the goal model is sensitive to ASR errors. The **Confirm** system avoids this drop in recall (the difference with the baseline is not statistically significant) and provides a larger gain in precision (5.2%), closely approximating the oracle system.

Table 1: *Concept understanding precision and recall.*

Condition	WER %	Precision %	Recall %
Baseline	32.8	58.0	63.9
Oracle	32.7	64.5	63.2
NoConfirm	33.9	62.0	60.4
Confirm	32.7	63.2	63.1

Overall, these results indicate that the main contribution of our model is to reduce the number of false positives by giving low probabilities to concept values that are inconsistent with the expressed user goal.

5. Discussion

By relying on representations and data structures that already exist in a dialog manager, such as the user goal structured into concept-value pairs, our approach can be easily integrated in many existing systems. The only additional human effort is that of designing the user goal BN, i.e. to capture natural dependencies between concepts. Once the structure is given, all the parameters of the dynamic SLM (including n-grams, intra-class probabilities, and the goal model's CPDs) can be learned from annotated dialogs. In the case of systems such as [13] that already maintain a probabilistic belief over user goal, there is little additional computational cost to performing dynamic intra-class word probability update, provided the ASR allows such operation. However, it should be noted that the BNs described in the POMDP literature focus on integrating noisy observations over several turn to build an accurate belief, rather than on exploiting prior domain knowledge. Therefore, those BNs have often low connectivity and the values of the CPDs are based on intuition rather than trained on data. In contrast, inter-concept dependencies and data-driven CPDs are crucial to the success of our approach because it predicts the value of a particular concept before it is uttered by the user.

Our evaluation has revealed that, at least in the Let's Go domain, the performance gain provided by the proposed model mainly comes from its penalization of inconsistent concept values, rather than in boosting likely concept values. In retrospect this is not surprising since it is easier to predict that the user will *not* provide an inconsistent concept value rather than *which* specific value they will provide. By adding more knowledge (i.e. nodes and arcs to the goal model), the model could provide more accurate concept value probabilities and therefore improve concept recall. This gain should be maximized in systems that feature a richer context than Let's Go, such as embodied agents. Finally, the relative low performance of the **NoConfirm** condition indicates that the model and the way the evidence is added is not sufficiently robust to ASR errors. An important direction of research is to improve the robustness of our approach by incorporating more information from the recognizer such as confidence scores and n-best lists, as has been proposed by Thomson et al [13] and others.

[2]Significance tests of precision and recall were done at the 0.05 level using Student's t-test with each dialog taken as an independent sample.

6. Conclusion

In this paper, we introduced a new framework for Statistical Language Modeling in spoken dialog systems that combines known techniques such as class-based SLMs and Bayesian Network-based user goal tracking. Intra-class word probabilities are updated using a goal model that captures inter-concept dependencies. We confirmed the effectiveness of the approach on a corpus of dialogs from the Let's Go system, a publicly deployed bus information system.

7. References

[1] Popovici, C. and Baggia, P., "Specialized Language Models Using Dialogue Predictions", Proc. ICASSP'97, Munich, Germany, 1997.

[2] Xu, W. and Rudnicky, A., "Language Modeling for Dialog Systems", Proc. ICSLP 2000, Beijing, China, 2000.

[3] Wessel, F. and Baader, A., "Robust Dialogue-State Dependent Language Modeling using Leaving-One-Out", Proc. ICASSP'99, Phoenix, AZ, USA, 1999.

[4] Lane, I. R. and Kawahara, T. and Matsui, T. and Nakamura, S., "Dialogue Speech Recognition by Combining Hierarchical Topic Classification and Language Model Switching", IEICE Trans. Inf. Syst., E88-D:3, pp. 446-454, 2005.

[5] Gildea, D. and Hofmann, T., "Topic-based language models using EM", Proc. 6th European Conference on Speech Communication and Technology(EUROSPEECH'99), Budapest, Hungary, 1999.

[6] Solsona, R. A. and Fosler-Lussier, E. and Kuo, H.-K. J. and Potamianos, A. and Zitouni, I., "Adaptive Language Models for Spoken Dialogue Systems", Proc. ICASSP'02, Orlando, FL, USA, 2002.

[7] Gruenstein, A. and Wang, C. and Seneff, S., "Context-Sensitive Statistical Language Modeling", Proc. Interspeech 2005, Lisbon, Portugal, 2005.

[8] Wiggers, P. and Rothkrantz, L. J. M., "Improving speech recognition by utilizing domain knowledge and confidence measures", in Lecture Notes in Artificial Intelligence 2807: Text, Speech and Dialogues 2003.

[9] Brown, P. F. and Della Pietra, V. J. and deSouza, P. V. and Mercer, R. L. "Class-based n-gram models of natural language." In Computational Linguistics 18(4), 1992, 467-479.

[10] Raux, A. and Bohus, D. and Langner, B. and Black A. W. and Eskenazi, M., "Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience", Proc. Interspeech 2006, Pittsburgh, USA, 2006.

[11] Ward, W. and Issar, S. "Recent improvements in the CMU spoken language understanding system.", Proc. ARPA Human Language Technology Workshop, pp. 213.216, 1994.

[12] Stolcke, A. "SRILM an Extensible Language Modeling Toolkit.", Proc. ICSLP'02, Denver, CO, USA, 2002.

[13] Thomson, B. and Young, S. "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems.", Computer Speech and Language, 2009.

[14] Sphinx 4, http://cmusphinx.sourceforge.net/sphinx4, 2010.

[15] Pearl, J. Probabilistic Reasoning in Intelligent Systems, San Mateo: Morgan Kaufmann, 1988.

[16] Intel, "Probabilistic Network Library", http://sourceforge.net/projects/openpnl, 2010.