Stochastic Bi-Languages to model Dialogs

M. Inés Torres
Dpto. Electricidad y Electrónica
Universidad del País Vasco. Bilbao, Spain
manes.torres@ehu.es

Abstract

Partially observable Markov decision Processes provide an excellent statistical framework to deal with spoken dialog systems that admits global optimization and deal with uncertainty of user goals. However its put in practice entails intractable problems that need efficient and suboptimal approaches. Alternatively some pattern recognition techniques have also been proposed. In this framework the joint probability distribution over some semantic language provided by the speech understanding system and the language of actions provided by the dialog manager need to be estimated. In this work we propose to model this joint probability distribution by stochastic regular *bi-languages* that have also been successfully proposed for machine translation purposes. To this end a Probabilistic Finite State Bi-Automaton is defined in the paper. As an extension to this model we also propose an attributed model that allows to deal with the task attribute values. Valued attributed are attached to the states in such a way that usual learning and smoothing techniques can be applied as shown in the paper. As far as we know it is the first approach based on stochastic bi-languages formally defined to deal with dialog tasks.

1 Introduction

Spoken Dialogue Systems (SDS) aim to enable people to interact with computers, using the spoken language in a natural way (Young, 2000; Raux et al., 2006). However, the management of an SDS is a very complex task that involves many other problems to be solved like the Automatic Speech Recognition (ASR), semantic representation and understanding, answer generation, etc. According to the information provided by the user and the history of previous dialogues the dialogue manager must decide the next action to be taken. Due to its complexity the design of dialogue managers has been traditionally related to rules based methodologies (Lee et al., 2006), sometimes combined with some statistical knowledge (Varges et al., 2009). These methodologies have been successfully used for specific tasks. But they are hard to be developed and they lack sensitivity to changes present in real tasks. Then plan-based and task-independent dialogue managers have been also proposed (Raux et al., 2006; Bohus and Rudnicky, 2009). In addition, statistical models based on Markov Decision Processes can be found for dialogue managers (Levin et al., 2000; Young, 2000).

Partially Observable Markov Decision Processes (POMDP) provided an excellent statistical framework that admits global optimization and deal with uncertainty of user goals (Williams and Young, 2007). This formulation is now considered as the state of the art of statistical SDSs. However, its put in practice entails intractable problems that need efficient and suboptimal approaches such as factorization of the state space and partition of the dialogue state distributions (Williams and Young, 2007; Williams, 2010; Lee and Eskenazi, 2012a; S. Young and Williams, 2013).

On the other hand new and challenging natural language processing tasks involving dialog are also arising in the last years. Some examples include the

Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, pages 9–17,
St Andrews–Sctotland, July 15–17, 2013. ©2013 Association for Computational Linguistics

analysis, or generation, of online debates (Walker et al., 2012) and the story generation for games using film dialog corpora as training sample (Lin and Walker, 2011).

In the pattern recognition framework statistical models have also been proposed to represent SDS (Georgila et al., 2006; Griol et al., 2008). Specifically in the interactive pattern recognition framework (Toselli et al., 2011; Torres et al., 2012) the joint probability distribution over some semantic language provided by the speech understanding system and the hypotheses provided by the dialog manager need to be estimated.

In this work we propose to model this joint probability distribution by stochastic regular *bi-languages*. These languages have also been successfully proposed to deal with machine translation (Torres and Casacuberta, 2011). To this end a Probabilistic Finite State Bi-Automaton is defined (PFSBA) in the paper. As an extension to this model we also propose an attributed model that allows to deal with the task attribute values. Valued attributed are attached to the states in such a way that usual learning and smoothing techniques can be applied as shown in the paper. As far as we know it is the first approach based on stochastic languages formally defined to deal with dialog tasks. Section 2 presents spoken dialog systems as a statistical, and interactive, pattern recognition problem.

Section 3 summarizes basic concepts related to stochastic regular *bi-languages* and then proposes a probabilistic finite-state *bi-automata* to model dialogs. We then extend this proposal to define an attributed model in Section 4. In Section 5 we deal with learning and smoothing procedures as well as with dialog generation and human-machine interaction tasks. Finally some concluding remarks are presented in Section 6

2 Spoken Dialog Systems

A classical pattern recognition system derives an hypothesis from some input stimulus, according to some previously obtained model. Let us now consider an SDS as an interactive pattern recognition system (Toselli et al., 2011). Let h be the an hypothesis or output that the dialog manager of an SDS proposes. Then the user provides some feedback signals, f, which iteratively help the dialog manager to refine or to improve its hypothesis until it is finally accepted by the user, as the diagram in Figure 1 a) shows. The hypotheses of the dialog manager are usually called *actions* in SDS literature. These actions typically consist in machine turns that include queries to a database to get the information required by the user, questions to the user to complete the data the system needs to fulfill user goals, strategies to recover recognition or understanding errors, turns providing information to the user as well as greeting turns.

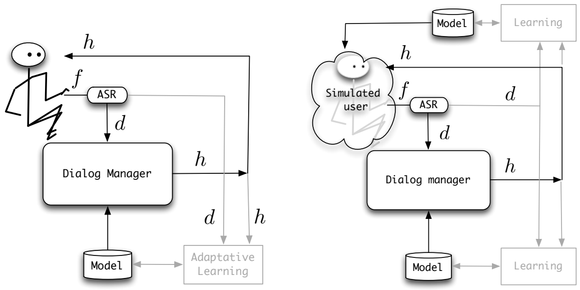

Figure 1: a) Diagram of an SDS where a Dialog Manager provides an hypothesis h given the previous hypothesis and a decoding d of a user feedback f. b) In the next interaction step a *Simulated User* provides the feedback f given its previous feedback and the system hypothesis h.

A basic simplification is to ignore the user feedback except for the last interaction or hypothesis h'. Assuming the classical *minimum-error criterion* the Baye's decision rule is simplified to maximize the posterior $P(h|h', f)$, and a *best* hypothesis $\hat{h}$ is obtained as follows:

$$\hat{h} = \arg\max_{h \in \mathcal{H}} P(h|h', f) \qquad (1)$$

This maximization procedure defines the way the dialog manager of an SDS chooses the best hypothesis in the space of hypotheses $\mathcal{H}$, i.e. the best action at each interaction step, given the previous hypothesis h' and the user feedback f. However, alternative criteria could also be considered to make this decision. In fact, the strategy of dialog managers is usually based on maximizing the probability of achieving the unknown user goals at the end of the interaction procedure while minimizing the cost of getting them (Williams and Young, 2007).

In an SDS, the interpretation of the user feedback f can not be considered a deterministic process. In fact the space of decoded feedbacks $\mathcal{D}$ is the output of an ASR system. Thus a *best* hypothesis can be obtained as follows (Toselli et al., 2011; Torres et al., 2012):

$$\begin{aligned}\hat{h} &= \arg\max_{h \in \mathcal{H}} \sum_{d \in \mathcal{D}} P(h, d|h', f) \\ &\approx \arg\max_{h \in \mathcal{H}} \max_{d \in \mathcal{D}} P(h|d, h')P(f|d)P(d|h')\end{aligned}$$

where f is the user turn, d is the decoding of the user turn, h is the hypothesis or the output produced by the system and h' is the *history of the dialog*.

A suboptimal approach can be considered through a two step decoding: find first an optimal user feedback $\hat{d}$ and then, use $\hat{d}$ to decode system hypothesis $\hat{h}$ as follows:

$$\hat{d} = \arg\max_{d \in \mathcal{D}} P(f|d)P(d|h') \quad (2)$$

$$\hat{h} \approx \arg\max_{h \in \mathcal{H}} P(h|\hat{d}, h') \quad (3)$$

Simulated User

The development of a complete SDS requires an online learning to train the dialog manager strategy. Therefore, a large amount of dialogues is needed together with real users with different goals, expectations and behaviors. Thus, statical dialog managers are usually trained by *simulated users* (Levin et al., 2000). A *simulated user* must provide the feedback f to the system at each interaction step. The user feedback f depends on its previous feedback f' according to some unknown distribution $P(f|f', h)$, which represents the user response to the history of system hypotheses and user feedbacks. This distribution considers the user behavior and stands for the user model $\mathcal{M}_{\mathcal{U}}$ and can also be defined considering now the user point of view. However, feedback f' produced by the user in the previous interaction is not corrupted by any noisy channel, such as an ASR system, before arriving to the user again. Thus, a deterministic decoding $d : \mathcal{F} \rightarrow \mathcal{D}$ maps each user turn signal into its corresponding unique decoding $d' = d(f')$ before arriving to the *user model*. Consequently the *best* decoded user feedback $\hat{d}$ is the one that maximizes the posterior $P_{\mathcal{M}_{\mathcal{U}}}(d|d', h)$

$$\hat{d} = \arg\max_{d \in \mathcal{D}} P(d|d', h) \approx \arg\max_{d \in \mathcal{D}} P_{\mathcal{M}_{\mathcal{U}}}(d|d', h) \quad (4)$$

where $\hat{d}$ is estimated using only the hypothesis produced by the system and the feedback produced by the user in the previous interaction step according to its *user model*. Figure 1 b) shows a *simulated user* interacting with a dialog manager according with a model of the *user behavior*. Equation (4) represents the way the user model decides the feedback to be produced at each interaction step. As in the case of the dialog manager, alternative criteria could be also considered to simulate the user behavior. In fact, many simulated user models can be found in the bibliography related to SDSs (Levin et al., 2000; Georgila et al., 2006; Lee and Eskenazi, 2012b). Figure 2 shows some user-manager interaction steps.

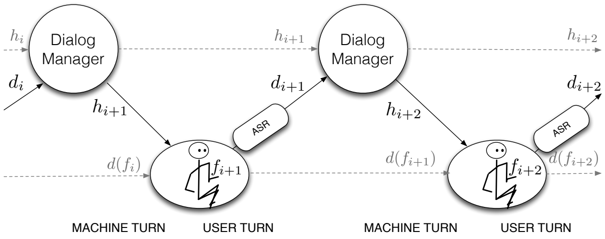

Figure 2: User-Manager interaction steps

3 Probabilistic Finite State Bi-Automata to model Dialogs

In this section we are defining a probabilistic finite-state model to deal with both the dialog manager hypothesis probability distribution $P(h|d, h')$ and the user feedback probability distribution $P(d|h, d')$. The model will be able to generate dialogs as alternative sequences of dialog manager hypothesis $\tilde{h}_i$ and decoding of user feedbacks $\tilde{d}_i$ as in Figure 2.

Some basic definitions

We first summarize the basic definitions of bi-string and stochastic regular bi-language provided by Torres and Casacuberta (2011). Let Σ and Δ be two finite alphabets and $\Sigma^{\leq m}$ and $\Delta^{\leq n}$, the finite sets of sequences of symbols in Σ and Δ of length up to m and n respectively. Let $\Gamma \subseteq (\Sigma^{\leq m} \times \Delta^{\leq n})$ be a finite alphabet (*extended alphabet*) consisting of pairs of strings, that we call *extended symbols*, $(s_1 \ldots s_i : t_1 \ldots t_j) \in \Gamma$ such that $s_1 \ldots s_i \in \Sigma^{\leq m}$ and $t_1 \ldots t_j \in \Delta^{\leq n}$ with $0 \leq i \leq m$ and $0 \leq j \leq n$.

Definition 3.1. *A bi-language is a set of strings over an extended alphabet Γ, i.e., a set of strings of the form $\mathbf{b} = b_1 \dots b_k$ such that $b_i \in \Gamma$ for $0 \leq i \leq k$. A string over an extended alphabet Γ will be called bi-string.*

Torres and Casacuberta (2011) defined a stochastic *bi-language* as follows:

Definition 3.2. *Given two finite alphabets Σ and Δ, a stochastic bi-language $\mathcal{B}$ is a probability distribution over Γ^* where $\Gamma \subseteq (\Sigma^{\leq m} \times \Delta^{\leq n})$, $m, n \geq 0$. Let $\mathbf{z} = z_1 \dots z_{|\mathbf{z}|}$ be a bi-string such that $z_i \in \Gamma$ for $1 \leq i \leq |\mathbf{z}|$. If $Pr_{\mathcal{B}}(\mathbf{z})$ denotes the probability of the bi-string $\mathbf{z}$ under the distribution $\mathcal{B}$ then $\sum_{\mathbf{z} \in \Gamma^*} Pr_{\mathcal{B}}(\mathbf{z}) = 1$.*

Model definition

Let Σ be the finite alphabet of semantic symbols provided by some speech understanding system. Thus, $\tilde{d}_i = d_1 \dots d_{|\tilde{d}_i|} \in \Sigma^{\leq m}$ represents the decoding of a user feedback f. Let now Δ be the finite alphabet of dialog acts that compose each of the hypotheses $\tilde{h}_i = h_1 \dots h_{|\tilde{h}_i|} \in \Delta^{\leq n}$ provided by the dialog manager. Let $\mathbf{z}$ be a *bi-string* over the extended alphabet $\Gamma \subseteq \Sigma^{\leq m} \times \Delta^{\leq n}$ such as $\mathbf{z} : \mathbf{z} = z_1 \dots z_{|\mathbf{z}|}$, $z_i = (\tilde{d}_i : \tilde{h}_i)$ where $\tilde{d}_i = d_1 \dots d_{|\tilde{d}_i|} \in \Sigma^{\leq m}$ and $\tilde{h}_i = h_1 \dots h_{|\tilde{h}_i|} \in \Delta^{\leq n}$. Extended symbols $(\tilde{d}_i : \tilde{h}_i) \in \Gamma$ have been obtained through some alignment between $\Sigma^{\leq m}$ and $\Delta^{\leq n}$, i.e. between pairs of user feedbacks decoding provided at a user turn and dialog manager hypotheses provided at the next machine turn.

Let us now define a Dialog Model $\mathcal{DM}$ as a Deterministic and Probabilistic Finite-State *Bi-Automaton* (Torres and Casacuberta, 2011) $\mathcal{DM} = (\Sigma, \Delta, \Gamma, Q, \delta, q_0, P_f, P)$ where

- Σ and Δ are two finite alphabets representing semantic symbols provided by the user and dialog acts provided by the dialog manager respectively, Γ is an extended alphabet such that $\Gamma \subseteq (\Sigma^{\leq m} \times \Delta^{\leq n})$, $m, n \geq 0$. ϵ represents the empty symbol for both alphabets, i.e., $\epsilon \in \Sigma$, $\epsilon \in \Delta$ and $(\tilde{\epsilon} : \tilde{\epsilon}) \in \Gamma$. To simplify let $\tilde{\epsilon}$ be ϵ.

- $Q = Q_{\mathcal{M}} \bigcup Q_{\mathcal{U}}$ is a finite set of states labelled by bi-strings $(\tilde{d}_i : \tilde{h}_i) \in \Gamma$. The set $Q_{\mathcal{M}}$ includes machine states before a machine turn providing an hypothesis and the set $Q_{\mathcal{U}}$ includes user states before providing a feedback.

- $\delta \subseteq Q \times \Gamma \times Q$ is the union of two sets of transitions $\delta = \delta_{\mathcal{M}} \bigcup \delta_{\mathcal{U}}$ as follows:
  - $\delta_{\mathcal{M}} \subseteq Q_{\mathcal{M}} \times \Gamma \times Q_{\mathcal{U}}$ is a set of transitions of the form $(q, (\epsilon : \tilde{h}_i), q')$ where $q \in Q_{\mathcal{M}}$, $q' \in Q_{\mathcal{U}}$ and $(\epsilon : \tilde{h}_i) \in \Gamma$
  - $\delta_{\mathcal{U}} \subseteq Q_{\mathcal{U}} \times \Gamma \times Q_{\mathcal{M}}$ is a set of transitions of the form $(q, (\tilde{d}_i : \epsilon), q')$ where $q \in Q_{\mathcal{U}}$, $q' \in Q_{\mathcal{M}}$ and $(\tilde{d}_i : \epsilon) \in \Gamma$

- $q_0 \in Q_{\mathcal{M}}$ is the unique initial state and it is labelled as $(\epsilon : \epsilon)$.

- $P_f : Q \rightarrow [0, 1]$ is the final-state probability distribution

- $P : \delta \rightarrow [0, 1]$ defines transition probability distributions ($P(q, b, q') \equiv Pr(q', b|q)$ for $b \in \Gamma$ and $q, q' \in Q$) such that:

$$P_f(q) + \sum_{b \in \Gamma, q' \in Q} P(q, b, q') = 1 \quad \forall q \in Q \quad (5)$$

  where a transition (q, b, q') is completely defined by q and b. Thus, $\forall q \in Q$, $\forall b \in \Gamma$ $|\{q' : (q, b, q')\}| \leq 1$

Let $\mathbf{z}$ be a *bi-string* over the extended alphabet $\Gamma \subseteq \Sigma^{\leq m} \times \Delta^{\leq n}$ such as $\mathbf{z} : \mathbf{z} = z_1 \dots z_{|\mathbf{z}|}$, $z_i = (\tilde{d}_i : \tilde{h}_i)$. $\mathbf{z}$ represents a dialog when z_i is of the form $z_i = (\epsilon : \tilde{h}_i)$ for machine turns m_i and $z_i = (\tilde{d}_i : \epsilon)$ for user turns u_i. Both, user and machine turns can also be null *bi-strings* of the form $(\epsilon : \epsilon)$. Let now $\theta = (q_0, z_1, q'_1, z_2, q_2, \dots, q'_{|\mathbf{z}|-1}, z_{|\mathbf{z}|}, q_{|\mathbf{z}|})$, $q_i \in Q_{\mathcal{M}}$, $q'_i \in Q_{\mathcal{U}}$, be a path for $\mathbf{z}$ in $\mathcal{DM}$. The probability of generating θ is:

$$Pr_{\mathcal{DM}}(\theta) = \left(\prod_{j=1}^{|\mathbf{z}|} P(q_{j-1}, z_j, q'_j) \right) \cdot P_f(q_{|\mathbf{z}|}) \quad (6)$$

$\mathcal{DM}$ is unambiguous. Then, a given *bi-string* $\mathbf{z}$ can only be generated by $\mathcal{DM}$ through a unique valid path $\theta(\mathbf{z})$. Thus, the probability of generating $\mathbf{z}$ with $\mathcal{DM}$ is $Pr_{\mathcal{DM}}(\mathbf{z}) = Pr_{\mathcal{DM}}(\theta(\mathbf{z}))$.

Figure 3 shows a $\mathcal{DM}$ where bold lines define path θ matching some *bi-string*.

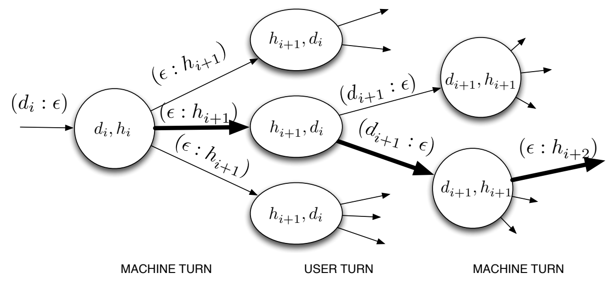

Figure 3: Probabilistic bi-automata $\mathcal{DM}$ defined in Section 3. Bold lines show a path θ matching some *bi-string*

Example Consider a simple railway information system where the alphabet of semantic units provided by the speech understanding system is defined as $\Sigma = \{Question, Sched, dest, orig, confirm\_y, confirm\_n, want, \ldots\}$ and the alphabet of dialog acts provided by the dialog manager is defined as $\Delta = \{Open, Close, Consult, Inf\_dest, Ask\_confirm, Ask\_day, \ldots\}$. Figure 4 shows a piece of a dialog labelled by sequences of symbols of Σ and Δ. This *dialog* is represented by the *bi-string* $\mathbf{z} = \{([\epsilon] : [Open])\ ([Question][Sched][dest] : [\epsilon])\ ([\epsilon] : [Consult\_sched][Inf\_dest][Ask\_confirm])\ ([Confirm\_y][want][orig] : [\epsilon])\ ([\epsilon] : [Ask\_day])\ ([Confirm\_y][day] : [\epsilon])\}$.

Machine: Welcome to the railway information system. Can I help you? *M:[Open]*
User: I would like some information on train schedules to Edinburgh. *U:[Question][Sched][dest]*
Machine: OK. I am looking for train schedules to Edinburgh. Is that ok? *M: [Consult_sched][Inf_dest] [Ask_confirm]*
User: Yes. I would like to travel from London. *U:[Confirm_y][want][orig]*
Machine: Are you traveling today? *M: [Ask_day]*
User: Yes, today *U: [Confirm_y][day]*
......

Figure 4: An example of a dialog in a railway information task. Machine turns are labelled as sequences $\tilde{h}_i \in \Delta^{\leq n}$ and User turns are labelled as sequences of decoding units $\tilde{d}_i \in \Sigma^{\leq m}$

Let us now consider a dialog model $\mathcal{DM}$ inferred from a training corpus consisting of dialogs of the railway information task such as the one shown in Figure 4. The sequence of states that correspond to the generation of the *bi-string* $\mathbf{z}$ is shown in the first column of Table 1. The second column shows the corresponding state labels as well as the next machine or user turn. The probability of *bi-string* $\mathbf{z}$ being generated by $\mathcal{DM}$ is calculated as $Pr_{\mathcal{DM}}(\mathbf{z}) = Pr_{\mathcal{DM}}(\theta(\mathbf{z}))$ according to Equation 6 where θ is the path for $\mathbf{z}$ in $\mathcal{DM}$, i.e. $\theta = (q_0, z_1, q_1, z_2, q_2, z_3, q_3, z_4, q_4, z_5, q_5, z_6, q_6, \ldots)$, $q_0, q_2, q_4, q_6 \in Q_{\mathcal{M}}$, $q_1, q_3, q_5 \in Q_{\mathcal{U}}$ as shown in Table 1, z_1, z_3, z_5 represent dialog manager hypotheses of the form $z_i = (\epsilon : \tilde{h}_i)$ and z_2, z_4, z_6 represent user feedback decodings of the form $z_i = (\tilde{d}_i : \epsilon)$.

4 Attributed model

The Dialog Model $\mathcal{DM}$ defined in Section 3 considers actions proposed by the dialog manager, i.e. sequences of dialog acts $\tilde{h}_i$ and sequences of decoding of user feedbacks $\tilde{d}_i$. Additionally, each machine and/or user state need to be labelled with the values of all relevant internal variables, which can be updated after each user turn. Thus, an additional alphabet appears to represent valued attributes of these internal variables, thus leading to an *attributed* model.

Attributed finite automata were proposed for syntactic pattern recognition in the nineties (Koski et al., 1995). The concept of attributed automata is a generalization of a finite automaton with attributes attached to states and contextual conditions as well as computational relations attached to transitions (Meriste, 1994). A transition is labelled with an input symbol, a context condition and an attribute evaluation function. Attributed automata specify context-sensitive languages. However in cases of finite domains of attributes an attributed automaton can be transformed to a finite automaton which simulates its external behavior, i.e. the attributed recognizer. This simulation is based on homomorphisms (Meriste, 1994). Stochastic versions of attributed grammar were then developed (Abney, 1997). However Abney (1997) showed that attribute-value grammars cannot be modeled adequately using statistical techniques which assume that statistical dependencies are accidental. More-

Table 1: The sequence of states along with their state labels that correspond to the generation of the *bi-string* $\mathbf{z} = \{([\epsilon] : [Open])\ ([Question][Sched][dest] : [\epsilon])\ ([\epsilon] : [Consult\_sched][Inf\_dest][Ask\_confirm])\ ([Confirm\_y][want][orig] : [\epsilon])\ ([\epsilon] : [Ask\_day])\ ([Confirm\_y][day] : [\epsilon])\}$. This *bi-string* represents the dialog in the Example

state	state label $(\tilde{d_i} : \tilde{h_i})$ and turn	state attributes
$q_0 \in Q_{\mathcal{M}}$	$([\epsilon] : [\epsilon])$ *M:[Open]*	none
$q_1 \in Q_{\mathcal{U}}$	$([\epsilon] : [Open])$ *U:[Question][Sched][dest]*	none
$q_2 \in Q_{\mathcal{M}}$	$([Question][Sched][dest] : [Open])$ *M: [Consult_sched][Inf_dest] [Ask_confirm]*	$Sched\_0.75\ dest\_0.25$
$q_3 \in Q_{\mathcal{U}}$	$([Question][Sched][dest] : [Consult\_sched][Inf\_dest][Ask\_confirm])$ *U:[Confirm_y][want][orig]*	$Sched\_0.75\ dest\_0.25$
$q_4 \in Q_{\mathcal{M}}$	$([Confirm\_y][want][orig] : [Consult\_sched][Inf\_dest][Ask\_confirm])$ *M: [Ask_day]*	$Sched\_c\ dest\_c\ orig\_0.75$
$q_5 \in Q_{\mathcal{U}}$	$([Confirm\_y][want][orig] : [Ask\_day])$ *U: [Confirm_y][day]*	$Sched\_c\ dest\_c\ orig\_0.75$
$q_6 \in Q_{\mathcal{M}}$	$([Confirm\_y][day] : [Ask\_day])$	$Sched\_c\ dest\_c\ orig\_c\ day\_0.50$

over, hard computational problems arise that need specific parser methods (Osborne, 2000; Malouf and van Noord, 2004).

In this section we propose an extension of the Dialog Model previously defined by adding a new alphabet of attributes. Assuming only discrete domains for the attributes they can be easily represented by a third string to be added to the state labels. Let us consider a dialog task characterized by a discrete set of internal variables such as *date, hour, time, dest, ...* These internal variables are a subset of the semantic decoding set, i.e. the subset of Σ set that consists of task dependent symbols. These internal variables can lead to simple $known, unknown$ attributes that can just be represented by the presence or absence of the attribute at each state. Thus, the new alphabet represents just the knowledge of the value as $\Omega = \{day, hour, time, dest, ...\}$. But we may desire a more detailed description of these values, such as *confirmed, known_to_some_extend, unknown,...* where *known_to_some_extend* can be quantified by a short set of confidence scores. Thus, the new alphabet is now $\Omega = \{day\_c, day\_0.75, day\_0.5, day\_0.25, hour\_c, hour\_0.75, hour\_0.5, hour\_0.25, ...\}$ where day_c means that the value of attribute day is confirmed and $day\_0.75$, $day\_0.5$ and $day\_0.25$ represent different confidence scores for the attribute value.

The model $\mathcal{DM}$ previously defined in Section 3 can now be extended to get an *attributed* model $\mathcal{A\_DM}$ by just adding another finite alphabet as follows $\mathcal{A\_DM} = (\Sigma, \Delta, \Gamma, \Omega, Q, \delta, q_0, P_f, P)$ where

- Σ and Δ are two finite alphabets representing semantic symbols provided by user and dialog acts provided by the dialog manager respectively, as defined in Section 3. Ω is a new finite set of symbols representing discrete valued attributes related with the dialog task.

- $Q = Q_{\mathcal{M}} \bigcup Q_{\mathcal{U}}$ is a finite set of states labelled by bi-strings $(\tilde{d_i} : \tilde{h_i}) \in \Gamma$ where now the valued attributes are also included $\tilde{w_i} \in \Omega^*$ as follows: $[(\tilde{d_i} : \tilde{h_i}), \tilde{w_i}]$

- $\delta \subseteq Q \times \Gamma \times Q$ is the union of two sets of transitions $\delta = \delta_{\mathcal{M}} \bigcup \delta_{\mathcal{U}}$, as defined in Section 3.

- $q_0 \in Q_{\mathcal{M}}$ is the unique initial state and it is labelled as $[(\epsilon : \epsilon), \epsilon]$

- $P_f : Q \rightarrow [0, 1]$ and $P : \delta \rightarrow [0, 1]$ define the final and transition probability distributions as before in Section 3.

The knowledge of the attributes leads to different strategies for the dialog manager. A score of 0.25 would lead to a *ask_confirm* dialog act whereas a confirmed value wouldn't need such a confirmation. Thus, the transition function $\delta \subseteq Q \times \Gamma \times Q$ and the transition probability distribution $P : \delta \rightarrow [0, 1]$ have a strong dependency of internal attributed attached to the states.

Example Let $\mathcal{A}\_\mathcal{DM}$ be an attributed model inferred from a training corpus consisting of dialogs of the railway information task such as the one shown in Figure 4 where the attribute alphabet has been defined as $\Omega = \{Sched\_c, ..., Sched\_0.25, dest\_c, ..., orig\_c, orig\_0.25, \dots\}$. Third column in Table 1 shows the attributes attached to each state of the path θ for the *bi-tring* $\mathbf{z}$ when generated by the $\mathcal{A}\_\mathcal{DM}$. The attributes first appear after the first user turn, which ask for schedules and inform about destination. But the ASR provides low confidence about the recognized values. Thus, the next machine hypothesis is an $Ask\_confirm$ dialog act. Finally machine state q_6 has attached two confirmed attributes and one more with some confidence score.

5 Putting models to work

In this section we deal with practical issues related to the use of the proposed models. We first deal with the learning and smoothing procedures. We then show two different application tasks: dialog generation and human-machine interaction.

Learning and smoothing models

Get a dialog corpus consisting of pairs of user and system turns. Then get the model topology as well as an initial maximum likelihood estimation of the parameters of both dialog manager and simulated user probability distributions. This model needs to deal also with unseen events, i.e. unknown situations at training corpus. The dialog manager can provide an hypothesis $(\epsilon : \tilde{h}_i)$ that does not lead to any of the existing states created when trained form the dialog corpus. In the same way simulated user can provide a user feedback $(\tilde{d}_i : \epsilon)$ not appearing in the training corpus, so not in the model. The model needs to be generalized in order to deal with any hypothesis or user feedback. Thus a back-off smoothing strategy needs to be carried out. This strategy creates new edges to existing nodes, when possible. Alternatively it searches for similar nodes to create the new edges (Torres et al., 2012). Thus the similarity between pairs of states needs to be estimated. States of the proposed models are labelled by $[(\tilde{d}_i : \tilde{h}_i), \tilde{w}_i]$, $(\tilde{d}_i : \tilde{h}_i) \in \Gamma \subseteq \Sigma^{\leq m} \times \Delta^{\leq n}$ and $\tilde{w}_i \in \Omega^*$. In practice, a string consisting of the concatenation of $\tilde{d}_i$, $\tilde{h}_i$ and $\tilde{w}_i$ is used. As a consequence string metrics like Levenshtein distance can be easily used to measure similarity between pairs of states (Georgila et al., 2006).

Generating dialogs: cooperative models

Figure 3 shows the way to generate dialogs with the proposed model. Let us train two models, one acts as a dialog manager and provides hypotheses according to Equation 3 and the other one acts as a simulated user according to Equation 4. However, different strategies can be defined in order to get a higher variability of dialogs. As an example Torres et al. (2012) proposes a random user feedback choice of decoded signals among the one defined in each state. Preliminary experiments were carried to generate a set of dialogs by training a model from dialog corpus representing real man machine dialogs aimed to get bus information systems. These experiments showed manageable model sizes, good task completion rates and good model behaviors.

On the other hand this formulation also suggests the joint decoding of user and machine turns. Find first a suboptimal best path $\hat{\theta}$ with a maximization approach and then estimate the best *bi-string* as follows:

$$\hat{\theta} \approx \arg\max_{\theta \in g(\mathbf{z})} Pr_{\mathcal{A}\_\mathcal{DM}}(\theta) \quad (7)$$

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} Pr_{\mathcal{A}\_\mathcal{DM}}(\hat{\theta}(\mathbf{z})) \quad (8)$$

where $g(\mathbf{z})$ denotes the set of possible paths in $\mathcal{A}\_\mathcal{DM}$ matching $\mathbf{z}$ and $\hat{\mathbf{z}} = z_1 \dots z_{|\mathbf{z}|}$, $z_i = (\tilde{d} : \tilde{h}_i)$ represents the best estimated dialog consisting of *bi-strings* z_i of the form $z_i = (\epsilon : \tilde{h}_i)$ for machine turns and on the form $z_i = (\tilde{d}_i : \epsilon)$ for user turns.

Human machine interaction

The proposed models can also be used to model SDS. The first model obtained from a dialog corpus would need to be improved by defining different dialog manager strategies and simulated user behaviors. Error recovery strategies need also be defined to deal with ASR errors. Then run the system until desired dialog goals are successfully achieved for different simulated user behaviors. Finally run the SDS with real users and use adaptive learning to obtain a dialog manager adapted to real interaction feedbacks.

6 Concluding remarks and future work

Some statistical pattern recognition techniques have been proposed to deal with spoken dialog systems. Specifically in the interactive pattern recognition framework the joint probability distribution over some semantic language provided by the speech understanding system and the language of actions provided by the dialog manager need to be estimated. In this work we have proposed to model this joint probability distribution by stochastic regular *bi-languages* that have also been successfully used for machine translation purposes. To this end a Probabilistic Finite State Bi-Automaton has been defined in the paper. As an extension to this model we have also proposed an attributed model that allows to deal with the task attribute values. Valued attributed are attached to the states in such a way that usual learning and smoothing techniques can be applied. As far as we know it is the first approach based on stochastic bi-languages formally defined to deal with dialog tasks.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science under grant TIN2011-28169-C05-04 and by the Basque Government under grant IT685-13.

References

Steven P. Abney. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23:597–618.

Dan Bohus and Alexander I. Rudnicky. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech and Language*, 23:332–361.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of Interspeech*.

David Griol, Lluís F. Hurtado, Encarna Segarra, and Emilio Sanchis. 2008. A statistical approach to spoken dialog systems design and evaluation. *Speech Communication*, 50:666–682.

Antti Koski, Martti Juhola, and Merik Meriste. 1995. Syntactic recognition of ecg signals by attributed finite automata. *Pattern Recognition*, 28:1927–1940.

Sungjin Lee and M. Eskenazi. 2012a. Pomdp-based let's go system for spoken dialog challenge. In *IEEE SLT Workshop*, pages 61–66.

Sungjin Lee and Maxine Eskenazi. 2012b. An unsupervised approach to user simulation: toward self-improving dialog systems. In *Proc. of SIGDIAL*, pages 50–59, Korea.

Cheongjae Lee, Sangkeun Jung, Jihyun Eun, Minwoo Jeong, and Gary Geunbae Lee. 2006. A situation-based dialogue management using dialogue examples. In *Proceedings of ICASSP*, pages 69–72.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transaction on Speech and Audio Processing*, 8(1):11–23.

Grace I. Lin and Marilyn A. Walker. 2011. All the world's a stage: Learning character models from film. In *Proceedings of the Conference on Artificial Intelligence and Digital Entertainment*.

Robert Malouf and Gertjan van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *Proceedings of the first International Conference on Natural Language Processing*, Hainan, China.

Merik Meriste. 1994. Attributed automata. some results and open problems. Technical report, Computer Science Department. Institute of Cybernetics, Estonian Academy of Sciences, December.

Miles Osborne. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proc. 18th International Conference on Computational Linguistics*, pages 586–592.

Antoine Raux, Dan Bohus, Brian Langner, Alan W Black, and Maxine Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of lets go! experience. In *Interspeech*, pages 65–68.

B. Thomson S. Young, M. Gasic and J. Williams. 2013. Pomdp-based statistical spoken dialogue systems: a review. *Proc IEEE*, to appear.

M. Inés Torres and F. Casacuberta. 2011. Stochastic k-tss bi-languages for machine translation. In *Proceedings of the 9th International Workshop on Finite State Models for Natural Language Processing (FSMNLP)*, pages 98–106, Blois (France). Association for Computational Linguistics.

M. Inés Torres, Jose Miguel Benedí, Raquel Justo, and Fabrizio Ghigi. 2012. Modeling spoken dialog systems under the interactive pattern recognition framework. In *SSPR&SPR. Lecture Notes on Computer Science*, pages 519–528.

Alejandro H. Toselli, Enrique Vidal, and Francisco Casacuberta, editors. 2011. *Multimodal Interactive Pattern Recognition and Applications*. Springer.

Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, Alexei V. Ivanov, and Pierluigi Roberti. 2009. Combining pomdps trained with user simulations and rule-based dialogue management in a spoken dialogue system. In *Proc. of the ACL-IJCNLP*, pages 41–44.

Marilyn A. Walker, Pranav Anand, Rob Abbott, Jean E. Fox Tree, Craig Martell, and Joseph King. 2012. That's your evidence?: Classifying stance in online political debate. *Decision Support Systems*, 53:719–729.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21:393–422.

J.D. Williams. 2010. Incremental partiten recombination for efficient tracking of multiple dialog states. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, USA.

Steve Young. 2000. Probabilistic methods in spoken dialogue systems. *Philosophical Trans of the Royal Society*, 1769(358):1389–1402.