Semantic graph clustering for POMDP-based spoken dialog systems

Florian Pinault and Fabrice Lefèvre

University of Avignon, LIA - CERI, France.

{florian.pinault,fabrice.lefevre}@univ-avignon.fr

Abstract

Dialog managers (DM) in spoken dialogue systems make decisions in highly uncertain conditions, due to errors from the speech recognition and spoken language understanding (SLU) modules. In this work a framework to interface efficient probabilistic modeling for both the SLU and the DM modules is described and investigated. Thorough representation of the user semantics is inferred by the SLU in the form of a graph of frames and, complemented with some contextual information, is mapped to a summary space in which a stochastic POMDP dialogue manager can perform planning of actions taking into account the uncertainty on the current dialogue state. Tractability is ensured by the use of an intermediate summary space. Also to reduce the development cost of SDS an approach based on clustering is proposed to automatically derive the master-summary mapping function. A preliminary implementation is presented in the MEDIA domain (touristic information and hotel booking) and tested with a simulated user.

Index Terms: dialogue systems, POMDP, semantics structures, graphs

1. Introduction

In recent years, a considerable amount of work have been carried out to move from theory to practical ground the idea of learning optimal strategies for spoken dialogue systems. It is now possible to train DM policies from data from real-world corpora (collected with already deployed systems or by means of a Wizard-of-Oz setup). But applying machine learning techniques to the issue of mimicking the human behavior generally entails to collect very large amount of data. For this reason efforts have been recently devoted to develop user simulators able to generate synthetic data with good characteristics [1, 2].

However whenever enough data can be made available with respect to the number of trainable parameters, the size of the models remains an issue. Even in the case of simple slot-filling problems the number of possible dialogue states can be massive and thus makes their enumeration intractable. Despite constant improvements in the training algorithm efficiency, compression of the dialogue state space still remains the only applicable solution as soon as several thousands states have to be accounted for: a function is elaborated which maps the initial (master) space to a compressed (summary) space. This latter is built so as to contain all and only the pertinent information for the decision-making process. Defining this mapping can be difficult and has not received yet a fully satisfactory answer. Although some propositions emerge. They mainly rely on a prior structuring of the master space, for instance grouping states into partitions as in the HIS model [3] or factorizing states in Bayesian networks [4], which facilitate the definition of the summary space by means of simple features. One limitation of these approaches anyhow is that they rely on a well-defined domain ontology to perform well, either to form partitions iteratively or to define the Bayesian network structure. In our work an attempt is made to define mapping functions which do not impose a rigid structure for the master space.

After mapping functions have been defined to convert master states to summary states, the master-summary POMDP framework also requires a final mechanism to transform the summary actions (broad dialogue acts such as inform, request etc) back to effective master actions (fully-specified dialogue acts such as inform(name="Ibis Montmartre",etc), request(location) etc). In general a heuristic is applied to find the most suitable piece of information upon which the summary action can be applied. In our system at each dialogue turn a set of rules is used to derive all the possible actions considering the current dialogue situation (with an associated score to manually guide the DM towards a desirable behavior). Then the POMDP, instead of choosing an action on its own, is used to score all the proposed master actions based on its evaluation of the best summary actions. Eventually a mixed policy is obtained from the weighted sum of the scores from the rules and the POMDP which allow to make a final decision in the master space. The effect of rule-based and POMDP decisions can thus be balanced and this allow to benefit from both approaches in an integrated way.

The main originalities of the developed approach can be summed up as:
- the SLU module is frame-based and stochastic. In our case it implies that no strong assumption is made during the semantic extraction process on the domain ontology. No hard-coded constraints (such as logical rules) are involved. So users can express themselves more naturally and nonetheless be understood in the context of the task. Of course at the cost of providing the DM with rather complex and never observed structured semantic information.
- a standard POMDP model is used in the summary space: as a comparison, in the HIS approach, belief update is performed in the master space but then the planning in the summary space is based only on a MDP model (with the state being defined by few features from the master dialogue state, such as the probability of the best hypothesis, number of matching items in the database, etc). In the proposed approach a belief tracking is performed in both the master and summary spaces.

This paper is organised as follows. Section 2 presents the semantic frame representation used as the SLU interface. In Section 3 a novel summary POMDP method is proposed. In Section 4, semantic graph clustering and n-best list clustering are presented, including specific distances definitions. Practical applications and results are finally analysed in Section 5 on a tourist information and hotel booking task.

2. Graphs of semantic frames

Dialogue managers in dialogue systems have to make decisions based on the available information about the user's goal. From the system point of view, the user's goal is a compound of specific pieces of semantic information gathered during the interaction process. Depending on the domain, this compound can be expressed through structures of various complexity: from simple flat (or parallel) slots to graphs of semantic frames.

Amongst the available semantic representations, the semantic frames [5] are probably the most suited to the task of interest here, mostly because of their ability to represent negotiation dialogs in addition to pure information seeking. Semantic frames are computational models describing common or abstract situations involving roles, the frame elements (FEs).

The FrameNet project [6] provides a large frame database for English. As no such resource exists for French, we elaborated a frame ontology to describe the semantic knowledge of the MEDIA domain. As an illustration this ontology is composed of 21 frames (LODGING, HOTEL, LOCATION, DATE etc) and 86 FEs (Lodging_type, Hotel_facility etc), based on a set of around 140 elementary concepts (day, month, city, payment-amount, currency etc). All are described by a set of manually defined patterns made of lexical units and conceptual units (frame and FE evoking words and concepts). The training data are semi-automatically annotated by a rule-based process. Pattern matching triggers the instantiations of frames and FEs which are composed using a set of logical rules. Composition may involve creation, modification or deletion of frame and FE instances. After the composition step a graph of frames is associated to each training utterance. This process is task-oriented and is progressively enriched with new rules to improve its accuracy. Once a reference annotation is established the models of a two-step stochastic frame annotation process [7] are trained and used afterwards in the system to annotate new utterances.

3. Summary POMDP

Following the idea of [8], the summary POMDP method consists in defining mapping functions from master spaces into summary spaces. A way to derive a fully-specified system action a_t from a summary action must be also defined.

Bold face notation will be used hereafter to distinguish between variables which are simple graphs (s_t, $u_t^\star$, a_t and o_t^i) and n-best lists of graphs ($\mathbf{o_t}$ and $\mathbf{b_t}$). The index t, time or turn number, will be dropped when not useful. All graphs considered here are frame graphs.

3.1. Master space

At the master level, which is the intentional level, the user generates an exact utterance $u_t^\star$ (unobserved), and the speech recognizer provides the system with n noisy versions of $u_t^\star$ along with some scores p_i as an n-best list:

$$\mathbf{o_t} = [(o_t^1, p_1), \ldots, (o_t^n, p_n)] \tag{1}$$

During a simulation the exact $u_t^\star$ can be known. In an annotated corpus, $u_t^\star$ is obtained from a reference annotation. In real conditions, $u_t^\star$ is not available. For perfect non-noisy speech recognition and understanding, $\mathbf{o_t} = [(u_t^\star, 1.0)]$.

From $u_t^\star$ (resp. $\mathbf{o_t}$), which depends on the current turn only, we define the cumulative state s_t (resp. $\mathbf{b_t}$) which depends on the full dialogue history. The update formulas, being essentially compositions of semantic structures, are considered parts of the SLU module and are only be briefly described here.

The master state s_t is the exact dialogue state. Each utterance $u_t^\star$ is accumulated to a unique frame graph using the state-update formula:

$$s_t = Update_s(s_{t-1}, u_t^\star, a_t) \tag{2}$$

$Update_s$ includes a FSM which is used to maintain a grounding state of each piece of information. Moreover, the database is used to include in s_t the number of venues matching the information of s_t. The same method used in the SLU module to compose semantic tuples into graphes at the turn level [7] is be applied for the update operation between turns.

The master belief $\mathbf{b_t}$ represents the uncertainty on the current state s_t. As for states, each observation $\mathbf{o_t}$ is accumulated into a unique m-best list of frame graphs using the belief-update formula:

$$\mathbf{b_t} = Update_{\mathbf{b}}(\mathbf{b_{t-1}}, \mathbf{o_t}, a_t) \tag{3}$$

It follows that $\mathbf{b_t}$ is an m-best list of graphs and can be written as:

$$\mathbf{b_t} = [(b_t^1, q_1), \ldots, (b_t^m, q_m)] \tag{4}$$

$Update_{\mathbf{b}}$ uses an approximation similar to the one presented in [3] to process the score of the n-best list. The update is performed as a cross product of the two lists (1) and (4): computing all $Update_s(b_t^i, o_t^j, a_t)$, associated with probability $q_i.p_j$. Then, identical graphs are removed and their weights summed, followed by a pruning and a re-normalisation step.

3.2. Summary space

Our summary POMDP uses two mapping functions M_s and M_o defining the summary state $\tilde{s}_t$ and observation $\tilde{o}_t$. They can be handcrafted as described in [9] or learned by classifiers (clustering, see *infra*).

$$\tilde{s}_t = M_s(s_t) \tag{5}$$

$$\tilde{o}_t = M_o(\mathbf{b_t}) \tag{6}$$

Note that the summary observation $\tilde{o}_t$ is *not* computed from the master observation $\mathbf{o_t}$, but from the master belief $\mathbf{b_t}$.

The summary POMDP is defined as a classic POMDP on the states $\tilde{s}_t$ and observations $\tilde{o}_t$. In this POMDP, a summary belief $\tilde{\mathbf{b}}_t$ is monitored and represents a true distribution over $\tilde{s}_t$. The summary belief update is performed using a complete probability model learned from a corpus (transition and observation probabilities).

3.3. From summary to master action

Not all system actions are possible at each turn depending on the current dialogue situation. Then some generic rules are used to generate the set of possible master actions.

The premises of the rules are clauses of logical connectors combining features extracted from the n-best list $\mathbf{b_t}$ (master belief), such as those described in [9]. The set of possible actions is data-driven and relies on the information available in the semantic structure of the master belief. To each master action is associated a summary action. Note that the action list could be easily constrained through the rules, whereby addressing the problem of VUI-completeness [10].

3.4. Policy merging

Adding scores to the rules defining the action list ensures a complete ordering of the action list in a very simplistic manner which allows to define an hand-crafted policy (referred to as *baseline* hereafter). The *baseline* score Q^0 for each master action does not depend on the belief and is designed using simple heuristics such as:

$$Q^0(\text{AskCity}) > Q^0(\text{AskContraints}) > Q^0(\text{AskDate}) \quad (7)$$

The summary POMDP policy π also provides scores but for summary actions (using the set of α-vectors [11] to approximate the Q-value function). These scores are transfered to the corresponding master actions (possibly several). As a result summary policies can be combined with the *baseline* policy through a linear policy mixer at the state-action value function level:

$$Q^{mixed}(\mathbf{b}, \tilde{\mathbf{b}}, a) = \lambda Q^{\pi}(\tilde{\mathbf{b}}, a) + (1-\lambda) Q^0(\mathbf{b}, a) \qquad (8)$$

4. Clustering

The design of the summary variables $\tilde{s} = M_s(s_t)$ and $\tilde{o} = M_o(\mathbf{b_t})$ is crucial to obtain an efficient compression of the master space (preserving the useful information to distinguish between dialogue situations requiring different system actions). In this work we propose to perform an unsupervised k-means clustering, as an alternative to the manual definition of features.

In this purpose, a distance between graphs (master states s_t) must be defined but also a distance between n-best lists of graphs (master beliefs $\mathbf{b_t}$). The notion of mean (or center) of a cluster is introduced too.

Let denote G the set of all graphs and B the set of m-best list of graphs for any $m \in \mathbb{N}$.

4.1. Graph edit distance

A widespread measure of similarity between graphs is the Graph Edit Distance (GED) [12], a generalisation of the string edit distance. The GED is defined as the shortest edition path to transform on graph into another, allowing 6 atomic editions: node and edge deletions, substitutions and insertions. We use the fast implementation of the GED as a binary linear programming problem, successfully used in [13] for molecule classification with small graphs (of size < 30).

Let denote d_1 the GED between two master states s_t.

4.2. The belief space $\mathbb{B}$

In order to perform a k-means clustering, it is necessary to define the mean of a cluster. In this purpose, we use a space $\mathbb{B}$ for which the mean is well-defined as it has an addition and a scalar product. Then we identify B to $\mathbb{B}$ and G to a subset of B.

Let $\mathbb{B}$ be the set of all probability distributions over G, which are nonzero only for a finite number m of graphs, for any $m \in \mathbb{N}$. The mean of elements of $\mathbb{B}$ is well-defined and belongs to $\mathbb{B}$.

The bijection between B and $\mathbb{B}$ can be expressed as follows: for any $\mathbf{b} \in B$, written as (4), the probability distribution associated to $\mathbf{b}$ is $P_{\mathbf{b}} \in \mathbb{B}$ such that

$$P_{\mathbf{b}}(g) = q_i \quad \text{if } \exists i / g = b^i \qquad (9)$$

$$P_{\mathbf{b}}(g) = 0 \quad \text{for any other } g \qquad (10)$$

From this bijection, it follows that any $\mathbf{b} \in B$ has an associated $P_{\mathbf{b}} \in \mathbb{B}$, therefore the mean of a cluster of n-best lists of graphs is well-defined.

To embbed G into $\mathbb{B}$, we use δ the canonical injection associating any graph g to δ_g, the Dirac distribution at the point g, which is the n-best list with only one element of score 1:

$$g \mapsto \delta_g = [(g, 1.0)] \qquad (11)$$

4.3. n-best list distance

It is then possible to define a distance d_2 on the space $\mathbb{B}$:

$$d_2(\mathbf{b}, \mathbf{b}') = \inf_{X \sim P_{\mathbf{b}} \text{ and } X' \sim P_{\mathbf{b}'}} \mathbb{E}(d_1(X, X')) \qquad (12)$$

with $\mathbb{E}$ the expectation. $X \sim P_{\mathbf{b}}$ denotes a random variable X which follows a probability distribution $P_{\mathbf{b}}$.

As the distance d_2 is too hard to compute directly, we approximate d_2 with the measure of similarity d_3, assuming the independence between X and X', defined as: (13), assuming independence between X and X':

$$d_3(\mathbf{b}, \mathbf{b}') = \mathbb{E}(d_1(X, X')) = \sum_{i,j} q_i q'_j d_1(b^i, b'^j) \qquad (13)$$

with $\mathbf{b}$ and $\mathbf{b}'$ are two beliefs, written as in (4).

If d_2 is a genuine distance (the proof is too long to be given here), d_3 is *not* a distance. But it has the advantage of being a linear function. Thus the "distance" d_3 to a cluster mean is the average of the "distances" d_3 to each point of this cluster.

5. Experiments and results

The task considered in our experiments is the MEDIA task which consists in informing about the prices of hotels with some constraints on facilities and making a reservation if any eligible venue is found. About 100k different user goals are possible involving 13 binary slots for facilities and 3 non-binary slots (location, price and date).

5.1. Policies

As described in Section 3, the summary variables $\tilde{s}_t$ and $\tilde{o}_t$ are extracted from the exact master state s_t and the observed n-best list $\mathbf{b_t}$ (master belief) using the mapping functions M_s and M_o. Two different configurations of POMDP systems are evaluated differing by the mapping functions.

5.1.1. POMDP with hand-crafted summary: POMDP-HCsum

In the POMDP-HCsum, M_s is factored into two features: $M_s(s) = (rules(s), db(s))$. The rule-based category $rules(s)$ can take 9 values. The number of database matches $db(s)$ can be 0, 1 or many.

M_o uses the same features applied to its first best hypothesis and possesses an additional feature: $M_o(\mathbf{b}_t) = (rules(b^1), db(b^1), entr(\mathbf{b}_t))$. The entropy feature $entr(\mathbf{b}_t)$ is binary (high/low).

Removing states never encountered in the corpus, the final summary system has only 18 states and 36 observations (and 8 actions).

5.1.2. POMDP with clustered summary: POMDP-clusterSum

In the *POMDP-clusterSum* system, $M_s(s_t)$ and $M_o(\mathbf{b}_t)$ are defined by an unsupervised k-means clustering using the distances d_1 and d_3 defined in Section 4. The data consist in 10k graphs from simulated dialogues with the *baseline* system.

Costs associated to edition operations are chosen in order to emphasize, in this order: the information from the database, the grounding states and some frame/FE known to be relevant for the tasks. This human decision can be avoided by using uniform edition costs.

Due to the intense computation cost of linear programming for the GED, only 14 state clusters and 10 observation clusters have been used (still with 8 actions).

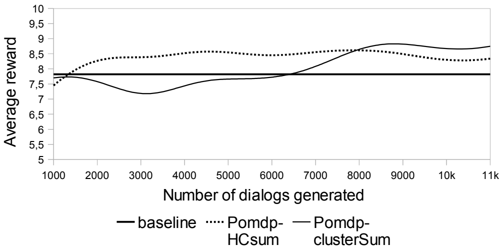

Figure 1: *Average reward during training epochs.*

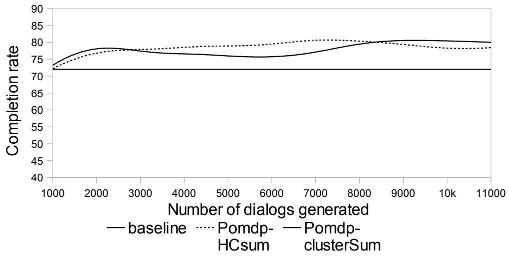

Figure 2: *Task completion rate during training epochs.*

5.2. Policy Training

The summary POMDP policies are optimized iteratively from an initial corpus C_0 created using the *baseline* policy:
1 - Collect transition and observation conditional probability tables from corpus C_i.
2 - Learn optimal policy π_i with model-based Perseus [11].
3 - Generate new data using an ϵ-greedy policy π_i (mixed with *baseline*).
4 - Merge new data with corpus C_i to create corpus C_{i+1}.
Iterations are repeated until the observed reward function reaches an asymptote.

At each iteration, 1000 dialogs are generated using the agenda-based user simulator from [2] on which the slot-based representation have been replaced by a semantic frame-based representation. Exploration coefficient (ϵ) is set to 0.1. Rewards are -1 at each dialogue turn and +20 for a final success. λ in policy mixer is set to balance the score dynamics between the *baseline* and the POMDPs so as to put them on a par.

5.3. Evaluation

Evolution of the average reward and completion rate along training epochs are presented in Figure 1 and 2. Iteration step is 1000 dialogues.

It can be observed that both POMDP policies give a clear improvement over the *baseline* alone in terms of average reward and task completion rate. Both policies seem to have comparable behavior even though on the first iterations, the handcrafted *POMDP-HCsum* performs slightly better than the automatic *POMDP-clusterSum*. A possible reason is the influence of the initial corpus generated with the *baseline* policy and thus more coherent with the design choices of the HC summary space.

Table 1 is populated with task completion rates and average rewards measured on 10k new dialogues. The two policies used are those obtained after 10 iterations (trained with 11000 generated dialogues). The results confirm the interest of using summary POMDPs for dialogue systems (gains are around +1 on rewards 7-8% on completion rates) and show that automatic compression of master space is an pertinent alternative to costly and complex expert design.

Policy	Task completion rate	Average reward
baseline	72.0	7.8
POMDP-HCsum	78.9	8.7
POMDP-clusterSum	80.3	9.2

Table 1: *Evaluation on 10k dialogues.*

6. Conclusion

This paper has described a way to interface a rich semantic representation with a POMDP-based dialogue manager. The manager is mixing ruled-based and POMDP policies. The summary POMDP is based on n-best lists of observations using either hand-crafted mapping functions or automatically derived functions from clustering of the frame graphs with appropriate distances. Experiments with a simulated user showed a performance improvement when using a summary POMDP compared to using only a manually defined policy, and the policy based on automatic compression of the dialogue state performs as well as one based on a hand-crafted summarisation. Evaluation with real users are in progress, and the preliminary results (with 20 users) tend to confirm the results with simulated users.

7. References

[1] K. Georgila, J. Henderson, and O. Lemon, “Learning User Simulations for Information State Update Dialogue Systems,” in *Eurospeech*, 2005.

[2] J. Schatzmann and S. Young, “The hidden agenda user simulation model,” *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 4, pp. 733–747, 2009.

[3] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, “The hidden information state model: A practical framework for POMDP-based spoken dialogue management,” *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, April 2010.

[4] B. Thomson and S. Young, “Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems,” in *Computer Speech and Language*, vol. 24, no. 4, 2010, pp. 562–588.

[5] J. Lowe, C. Baker, and C. Fillmore, “A frame-semantic approach to semantic annotation,” in *SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[6] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, “Background to framenet,” *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[7] M. Meurs, F. Lefèvre, and R. De Mori, “Spoken language interpretation: On the use of dynamic bayesian networks for semantic composition,” in *ICASSP*, 2009.

[8] J. Williams and S. Young, “Scaling up pomdps for dialog management: The ”summary pomdp” method,” in *Automatic Speech Recognition and Understanding*, 2005, pp. 177 –182.

[9] F. Pinault, F. Lefèvre, and R. De Mori, “Feature-based summary spaces for stochastic dialogue modeling with hierarchical semantic frames,” in *Interspeech*, 2009.

[10] R. Pieraccini, D. Suendermann, K. Dayanidhi, and J. Liscombe, “Are we there yet? Research in commercial spoken dialog systems,” in *Text, Speech and Dialogue TSD*, 2009.

[11] M. Spaan and N. Vlassis, “Perseus: Randomized point-based value iteration for pomdps,” in *JAIR*, 2005.

[12] X. Gao, B. Xiao, D. Tao, and X. Li, “A survey of graph edit distance,” *Pattern Analysis and Applications*, vol. 13, no. 1, pp. 113–129, February 2010.

[13] D. Justice and A. Hero, “A binary linear programming formulation of the graph edit distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1200 –1214, 2006.