Feature-based Summary Space for Stochastic Dialogue Modeling with Hierarchical Semantic Frames

Florian Pinault, Fabrice Lefèvre and Renato De Mori*
LIA - University of Avignon, BP 1228, 84911 Avignon, FRANCE
{florian.pinault, fabrice.lefevre, renato.demori}@univ-avignon.fr

Abstract

In a spoken dialogue system, the dialogue manager needs to make decisions in a highly noisy environment, mainly due to speech recognition and understanding errors. This work addresses this issue by proposing a framework to interface efficient probabilistic modeling for both the spoken language understanding module and the dialogue management module. First hierarchical semantic frames are inferred and composed so as to build a thorough representation of the user’s utterance semantics. Then this representation is mapped into a feature-based summary space in which is defined the set of dialogue states used by the stochastic dialogue manager, based on the partially observable Markov decision process (POMDP) paradigm. This allows a planning of the dialogue course taking into account the uncertainty on the current dialogue state and tractability is ensured by the use of an intermediate summary space.

A preliminary implementation of such a system is presented on the MEDIA domain. The task is touristic information and hotel booking, and the availability of WoZ data allows to consider a model-based approach to the POMDP dialogue manager.

Index Terms: dialogue modeling, spoken language understanding, POMDP, semantic frames

1. Introduction

Human-computer interaction in natural language is controlled by a dialogue manager (DM) that maintains a computational representation of the belief about user intentions and decides appropriate actions in order to perform a required task. The choice of the action to be performed by the computer system in each dialogue turn should be driven by a policy whose main objective it to effectively achieve a user goal. In the case of spoken dialogues, such a policy has to select the most appropriate actions in spite of the uncertainty due to the fact that speech input processed by error prone automatic speech recognition (ASR) system and spoken language understanding (SLU) is performed under the control of imperfect and incomplete knowledge.

Machine learning methods have been recently considered for learning models of DM strategies. Partially Observable Markov Decision Processes (POMDP) [1, 2] can be used for choosing actions to be performed by DMs. Though POMDP policy optimisation is known to be intractable for real-world tasks, some algorithms finding sub-optimal policies have been recently proposed. They can either be model-free as in [2] or be model-based as in [3, 4]. Model-free approaches do not require to explicitly model the transition and observation probabilities in POMDPs and estimate their parameters by a direct interaction with the environment. Model parameters could be estimated by analyzing a large number of human-machine dialogues with a large variety of users. As these data are seldom available, another possibility is to generate them with simulators. In such a case, a substantial effort has to be made to obtain a variety of realistic behaviours otherwise the system would just reproduce the simulator limits. A new possibility, introduced in this paper, consists in determining dialogue policies based on annotated dialogues between humans and a Wizard of Oz (WoZ) and in evaluating the resulting policies with simulation.

According to the literature, dialogue state modeling approaches can be classified into two broad types. In the case of large domain tasks, word patterns in the speakers’ utterance transcriptions are directly taken into account and used to trigger pre-defined actions. On the contrary, when the domain is small enough, dialogue states can be modeled by a database request filled with the information progressively provided by the user (slot-filling dialogue state, see for instance [1]). The approach proposed in this paper is inspired by the consideration that the state of complex dialogues, as those performing negotiations, should be described by structured knowledge representations like the ones obtained with computational structures such as frames. Recently, hierarchical frame semantic representations for spoken language understanding (SLU), based on the FrameNet paradigm [5] has been proposed [6].

A frame is a computational model for representing semantic entities and their properties. It is based on a data structure containing slots and fillers (which can be frames themselves). Frame instantiation may be triggered by evidence from the elements of the surface representation or by expectation. An SLU system described in [6] performs interpretation as a classification process. Classifiers automatically trained are used to obtain a surface semantic representation which elements are concept tags. Concept tags may represent relation names, variable types and values, or function names. In a second step, composition of tags is performed into hierarchical frame structures.

Composition knowledge can either be learned from examples or be compiled by experts. An interesting point is that, based on this representation, some assertions not present in a natural language message can be obtained by inference. SLU and DM progressively build frame instances that represent user intentions (and possibly remove some). A user request can be satisfied when it is enough specified to allow a consequent system action to be performed. In the approach described in this paper, semantic features are selected from a human expert and describe situations represented by sequences of system-user dialogue acts, amount of slots filled for instances of frames that characterize the focus of the conversation and others.

This paper is organized as follows. First the next section recalls the POMDP dialogue manager basic principles. Then the
semantic representation proposed as front-end to the dialogue manager is described in Section 3. The feature-based summary space is introduced in Section 4 with the details on how it is integrated in the POMDP model in Section 5. Finally experiments and results are reported in Section 6.

2. POMDP dialogue manager

An introduction to Markov decision process and reinforcement learning can be found in Sutton and Barto’s book [7]. A comprehensive introduction to POMDP is [8] and [2] gives a presentation of state-of-the-art POMDP use for dialogue management.

To cast dialogue management into the POMDP framework, one needs to define appropriate dialogue states $s_t$ and machine actions $a_t$ at each time step $t$. State and action spaces are assumed to be finite. Dialogue states represent the user’s goal and dialogue history. Any relevant information for the machine to make a decision should be included.

In a POMDP, a dialogue is supposed to be in any possible state at a given dialogue turn with a given probability for being in each state. The action to be performed in that turn is decided on the criterion for obtaining the maximum reward on the entire dialogue. In this purpose, a reward function $R$ is defined to compute the immediate reward $r_t = R(s_t, a_t, s_{t-1})$.

As the real dialogue state is considered unknown (“hidden”), only an observation $o_t$ of it is attainable. The belief vector $b_t$ is a sufficient statistic for the dialogue history and is computed from $b_{t-1}$ with the Bellman’s Bayesian update formula:

$$b_t(s) = \alpha P(o_t|s_t, a_{t-1}) \sum_s P(s_t|s, a_{t-1}) b_{t-1}(s)$$  \hspace{1cm} (1)

where $\alpha$ is a normalization constant. State transitions are characterized by a probability distribution $P(s_t|s_{t-1}, a_{t-1})$. And as dialogue states cannot be directly observed, relations between observations and states are summarized in the observation probability distribution $P(o_t|s_t, a_{t-1})$.

Solving the POMDP is computing a policy $\pi(b)$ which maximizes the value function $V^*(b)$ recursively defined by:

$$V^*(b) = E(r_t + \gamma.V^*(b_{t+1})|b_t = b, a_t = \pi(b))$$  \hspace{1cm} (2)

where $\gamma$ is a discount factor (usually 0.95) to lower the influence of far events.

The training algorithms iteratively compute the best value for $V$, w.r.t. the selected actions providing an optimal policy solution. Unfortunately the algorithms have a high computational complexity. Sub-optimal variants have been proposed but they can hardly handle more than few tens of observations, states and actions. This requires the summarization of the dialogue states in a small number of descriptors where it is assumed that machine actions can be decided from the information still available in this summary state. The resulting POMDP becomes more tractable due to the lower search space size.

In a previous work [9], an unsupervised procedure was proposed to elaborate the summary state space. However, although attractive from a cost perspective, this technique does not allow a precise and explainable definition of the summary states entailing a great difficulty in the setup of the reward function. Here we propose to derive the summary states from a semantic frame annotation of the users’ turns.

3. Hierarchical frame representation

This section outlines the understanding module implemented to convert raw acoustic data into hypotheses of word sequences, dialogue acts (DA), concept sequences and semantic frames. These annotations are used to define the states $s_t$, observations $o_t$ and actions $a_t$ used in the transition probabilities:

$$T = P(s_t|s_{t-1}, a_{t-1})$$  \hspace{1cm} and \hspace{1cm}  \hspace{1cm}$$Z = P(o_t|s_t, a_{t-1})$$

used in the POMDP model.

Table 1: Simplified example of semantic frame annotation in the MEDIA corpus.

<table>
<thead>
<tr>
<th>words</th>
<th>I'd like er an hotel er in Nice</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts</td>
<td>want, hotel, localization-hotel(Nice)</td>
</tr>
<tr>
<td>frames</td>
<td>F1: Hotel(city = 'Nice', name = ' ')</td>
</tr>
<tr>
<td></td>
<td>F2: Want(theme = 'F1')</td>
</tr>
</tbody>
</table>

MEDIA corpus

MEDIA is a corpus of dialogues in the domain of tourist information and hotel reservation [10]. The MEDIA corpus consists of 1257 dialogues (18,831 user utterances) recorded with a WoZ technique (a human simulating an automatic phone server). The MEDIA data have been manually and automatically transcribed. The word error rate on the MEDIA official test set is 33.5%.

Dialogue act tagging

The WoZ turns of the corpus have been annotated in terms of dialogue acts (DA) tags. Generally, this task can be avoided by logging appropriate information during the recording of the corpus, unfortunately it has not been done for MEDIA. The tagging is rule-based. Being generated by a human, the WoZ interactions do not always express as a unique DA. For instance, most database queries are combined with a final confirmation question. When such DA combination has many occurrences, compound DAs have been retained and used in the tagging. A total of 15 DAs are considered: 10 basic and 5 compound acts.

Concept annotation

A SLU module has been applied on the transcriptions to generate flat concept annotations. The semantic dictionary includes 83 basic concepts. A mode tag is added to the concepts, taking 4 different values (positive/negative/interrogative/optional). Semantic annotation includes values associated with concept tags.

The SLU module used in the experiments is based on Dynamic Bayesian Networks (DBN) [11]. The concept error rates are 21.3% on perfect transcriptions and 43.4% on ASR hypotheses.

Frame annotation

The frame annotation module uses the concept tag hypotheses and groups them into global frame semantic structures. The choice of a frame annotation in this work is motivated by its ability to represent negotiation dialogues and also to adapt to complex actions of the DM. A frame describes a common or abstract situation involving roles called frame elements (FE). For a given frame, the frame-evoking words and concepts are its lexical (LU) and conceptual (CU) units. A frame knowledge source (KS) has been manually defined to describe the semantic composition knowledge of the MEDIA domain. The MEDIA KS contains 21 frames and 86 FE.

Frames and FEs are described by a set of manually defined

<table>
<thead>
<tr>
<th>words</th>
<th>I'd like er an hotel er in Nice</th>
</tr>
</thead>
<tbody>
<tr>
<td>concepts</td>
<td>want, hotel, localization-hotel(Nice)</td>
</tr>
<tr>
<td>frames</td>
<td>F1: Hotel(city = 'Nice', name = ' ')</td>
</tr>
<tr>
<td></td>
<td>F2: Want(theme = 'F1')</td>
</tr>
</tbody>
</table>
Table 2: Examples of summary features. \( u_t \) (resp. \( w_t \)) is the set of frames in the user’s (resp. WoZ’s) utterance at time \( t \).

<table>
<thead>
<tr>
<th>State features</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_{\text{Query},u} )</td>
<td>( \text{card}{\text{frame}_{\text{Query}} \in u_t} )</td>
</tr>
<tr>
<td>( F_{\text{Hotel},u} )</td>
<td>( \text{card}{\text{non-empty frame}_{\text{Hotel}} \in u_t} )</td>
</tr>
<tr>
<td>( F_{\text{Query}} )</td>
<td>( \theta_1 F_{\text{Query},u} + \theta_2 F_{\text{Request},u} + \ldots + \theta_t F_{\text{Number},u} ) (( \theta_1 = +5, \theta_2 = -3, \ldots, \theta_t = +1 ))</td>
</tr>
<tr>
<td>( F_{\text{RecentS}} )</td>
<td>( (s_{t-1} = 3) \lor (s_{t-2} = 3) )</td>
</tr>
</tbody>
</table>

...  

patterns. These patterns are made of LUs and CUs. Some of the CUs match the MEDIA basic concepts, some others are defined according to the K5 frames. An example of the MEDIA frames Hotel and Want with three FEs named city, name and theme is given in Table 1.

In order to obtain frame annotations on the speech data, a two-step rule-based annotation process has been carried out: first conceptual and lexical units associated to frames are used to trigger new frames and their FEs when they match with concept or word inputs (in this order), then a set of logical rules is applied to compose these frames and infer new ones. In this latter step, the frames and FEs produced in the first step determine the truth values of the logical rules. According to these truth values, new frames and FEs can be created and current frames and FEs can be deleted, modified or connected (for instance some frames can be subframes of others, in this case they are connected through a FE taking a frame as value). DBNs have also been used for inferring frame annotations [6] and will be exploited in a future work.

### 4. Feature-based summary space

At each time step \( t \) (i.e. each speaker utterance), a dialogue state \( s_t \) is defined as a function of the dialogue history \( h_t \): \( s_t = M(h_t) \). The purpose of the mapping function \( M \) is to determine the next dialogue action. The function uses only the information required for this purpose and \( s_t \) is represented by a summary state \( \tilde{s}_t \). We propose to represent \( h_t \) as a set of features and to base the mapping \( M \) to the summary space on these features.

The dialogue history \( h_t \) includes all information about the past course of the dialogue: frame annotations of previous user utterances, past WoZ actions and previous states. Primary features in \( h_t \) are the raw data on top of which secondary features are added. Additional feature types can be integer (e.g. \( F_{\text{Query},u} \) the number of occurrences of the \( \text{Query} \) frame in the current user turn), real (e.g. \( F_{\text{Query}} \)) or boolean (e.g. \( F_{\text{RecentS}} \)). A secondary feature \( F \) is defined from other \( F_i \) in several different ways:

- **Pre-defined function**: \( F = \text{card}\{F_i\} \)
- **Linear combination**: \( F = \sum \theta_i F_i \)
- **Inverse maximisation**: \( F = \text{arg max}_i (F_i) \)
- **Logical clauses**: \( F = (F_1 \lor F_2) \land F_3 \)

In this process of state feature definition, ad-hoc design has been performed by a human expert. Some general data mining tools have been used (clustering, classifiers...) to guide the expertise and further investigations will be made to obtain an unsupervised (or lightly supervised) feature selection procedure. Primary features include graphs of semantic frames and so \( h_t \) has virtually an unlimited size. Table 2 gives some examples of features. In the MEDIA corpus, \( h_t \) contains an average of 49 different frames (std.dev.= 31). For example the frame \text{Person} can have up to 34 occurrences in some dialogue histories.

In this framework, the summary state \( \tilde{s}_t \) itself can be seen as a special secondary feature \( F_{\text{Summary}} \) (and \( M \) simply consists in returning its value). \( F_{\text{Summary}} \) has been defined from a careful analysis of the frame annotation to provide a consistent tiling of the state space. To give an idea, some rules of the mapping are given in the last row of Table 2. The design of \( \tilde{s}_t \) make use of about 15 counting features similar to \( F_{\text{Query},u} \), uses also \( \text{card}\{h_t\} \) and some complex features. A set of 18 summary states is used in our experiments (i.e. \( F_{\text{Summary}} \) is an integer in \([0..17]\)). The POMDP policy optimisation (and its evaluation) is performed on the summary space (i.e. considering sequences of \( s_t \)).

### 5. Frame-based POMDP modeling

#### Observation definition

In our context, the observations \( o_t \) are noisy versions of the dialogue states \( s_t \). So the state definition rules are used to derive the observations from the semantic frame representation. The difference between \( o_t \) and \( s_t \) lies in the use of noisy data (vs. reference data). The reference data are defined from human text transcription and human concept annotation (even if they are neither error-free nor so well-defined since the inter-annotator agreement is not perfect).

Two levels of noisy data are considered:

- corrupted only by the noise from the automatic concept annotation system, applied to the reference text transcription (SLU);
- or using the text transcription provided by the speech recognizer (ASR+SLU).

In both cases, the same hand-crafted approach is used at the end to obtain the semantic frame annotation.

#### Action definition

Machine actions are defined semi-automatically using expert rules. The WoZ protocol used during the corpus collect implies a stereotypical sentence structure. Therefore the actions \( a_t \) are easier to define than \( s_t \) or \( o_t \). A word-pattern based set of rules was sufficient to define consistent actions based on the DA tagging of the WoZ turns. The action definitions are closely related to the speech act theory in linguistics, and to the DAMSL taxonomy [12]. A summary space is also considered for the action definition, the master action set is projected on a smaller set to make POMDP optimization tractable.

#### Model parameters

The POMDP experiments (belief monitoring, simulations and evaluation) have been performed at a summary level, i.e. the considered states, observations and actions are resp. \( \tilde{s} \), \( \tilde{o} \) and \( \tilde{a} \). Belief monitoring in the master space (as in [2]) is still possible. But as the system does not use n-best list inputs, the impact would be small despite an increased complexity.

The conditional probabilities for state transitions \( P(s_t|s_{t-1}, a_{t-1}) \) and observations \( P(o_t|s_t, a_{t-1}) \) are directly estimated from the corresponding triplet sequences. Since the corpus coverage is not large enough to include every single combination of the tuples \( (s_{t-1}, a_{t-1}, s_t) \) and \( (a_{t-1}, s_t, o_t) \), a factored language model is used with a generalized backoff procedure to produce the probability estimates. The SRILM toolkit [13] is used in our experiments.
Table 3: Average rewards for three configurations of dialogue policy. For POMDP versions, the horizon length is 6.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Average reward</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>POMDP SLU</td>
<td>2.88</td>
<td>4.49</td>
</tr>
<tr>
<td>POMDP ASR+SLU</td>
<td>2.71</td>
<td>4.48</td>
</tr>
<tr>
<td>WoZ</td>
<td>4.86</td>
<td>10.2</td>
</tr>
</tbody>
</table>

6. Experiments

To make its complexity lower, the POMDP optimization is performed with a grid-based approach (PBVI algorithm [3]) and the optimization search is limited to a finite horizon of length 6. For these preliminary experiments, the performance of a policy \( \pi \) is asserted by simulating 1000 dialogues on a summary space level: \( \hat{a} \) generated by the machine following the policy \( \pi \) and \( \hat{s}, \hat{o} \) generated by a simulated user following the POMDP dynamics (encompassed in \( \hat{T} \) and \( \hat{Z} \)).

The reward function penalizes each time step \((-1)\) and evaluates correct vs. incorrect dialogue terminations as follows:

1. \( R(\hat{a} = \text{Closing}, \hat{s} = S_{\text{Closing}}) = +10 \)
2. \( R(\hat{a} = \text{Closing}, \hat{s} \neq S_{\text{Closing}}) = -10 \)

To assert the impact of the errors made by the SLU system, the two POMDP configurations described above (SLU and ASR+SLU) are compared. In the corpus data, the DM policy is applied by a human being (WoZ) following the MEDIA experimental protocol. This WoZ policy is also evaluated with the same reward function. Table 3 shows that the system’s behaviour with ASR is comparable to the one with exact speech transcription: the system is pretty robust against ASR errors.

The third row in Table 3 gives the WoZ policy reward. Of course, the WoZ had access to an information even richer than what is in the master state \( s \) itself. Therefore it performs better than the summary POMDP policies and provides a real-world reference. On the other hand, the computed policies are much more certain to achieve their average reward, as shown by a considerably smaller standard deviation (the deviation of simulated dialogues can be reduced to zero by increasing their number, but all reported evaluations use the same amount of data).

Figure 1 shows the average reward of the POMDP policies trained on SLU only and ASR+SLU for different planning horizon values. It shows that the difference between SLU and ASR+SLU are small. The slight improvement of the average reward after horizon 2 shows that planning with higher horizons has a small impact on the policy performance. The reason could be a too coarse definition of the dialogue states which would prevent the POMDP model to take full advantage of the corpus data. A solution could lie in an intermediate approach between the unsupervised and feature-based dialogue state definitions where the unsupervised procedure, as in [9], is applied on top of deterministic rules to refine the states according to an objective criterion, such as the average reward.

7. Conclusion

In this paper a dialogue manager is presented where policy optimization takes into account the uncertainty due to speech understanding errors thanks to probabilistic models. Compared to similar approaches, the representation of the user’s utterances by means of semantic frames allows to infer dialogue states from a richer and more structured representation than a simple slot-filling model. A feature-based summary space allows to map the potentially infinite state space to a space with a manageable complexity.

Dialogue simulations according to the probabilistic models allowed to evaluate the derived policies. It is observed that the errors involved by the word and concept recognition processes have limited impact on the policy performance measured in term of average reward. A full prototype is under construction and user trials will be soon available for more extensive evaluations of the proposed approach.

8. References