UFAL Dialogue state update

The purpose of this document is to describe the principles and implementation of dialogue state update in spoken dialogue systems developed at UFAL, MFF UK, Czech Republic.

Filip Jurcicek
06/05/12

Table of Contents

Definition of a dialogue state

In a spoken dialogue system, the purpose of a dialogue state is to track the progress in a dialogue. In its simplest form, the dialogue state can be composed only of variables that represents the goal of the user. A good example of this approach is a tourist information domain where users can ask for details about restaurants, bars, and hotels, users can constraint their search by area, price range, stars, and the dialogue system can provide information about the address, the postcode, or the phone number for the selected venue. In this domain, the goal can be composed of the following variables:

venue_type	– defines what type of a venue the user is looking for
area	– desired location of the venue
price_range	– price range of the venue
near	– whether the requested venue should be near another venue
food_type	– type of food served at the venue if the venue type is a restaurant
stars	– number of stars of the venue if the venue type is a hotel

and every time a user informs about a new constraint, the information is stored. Sometimes these variables are called slots or concepts and their values attributes. In addition to the goal, the dialogue state can store information about what the user requested or wants to confirm, what the system already informed about or what the system confirmed.

The dialogue state update

The UFAL dialogue systems uses the concept of Information State Update, where the state is updated whenever new information is available, either from a user or the system itself or any other partner in the communication. Since the update is defined on the dialogue state, the process of updating the dialogue states is called the dialogue state update. The dialogue state update is best depicted using an influence diagram.

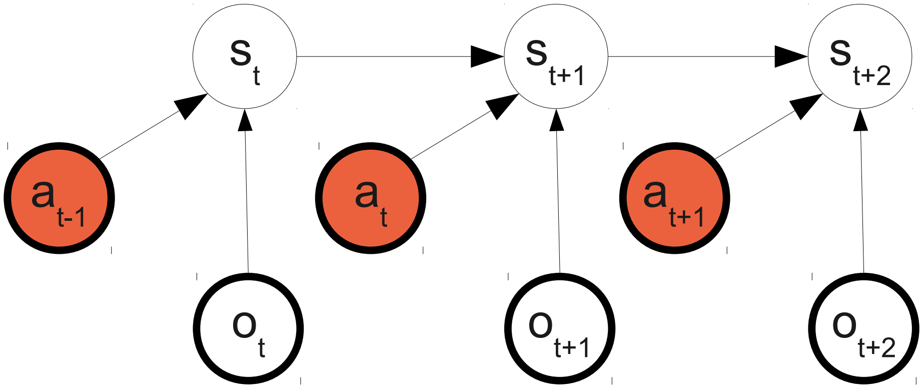

Figure 1: Dialogue state update influence diagram

Shown at Figure 1, the dialogues state $\mathbf{s_{t+1}}$ depends on the previous state $\mathbf{s_t}$, the previous system dialogue act $\mathbf{a_t}$, and the current user dialogue act $\mathbf{o_{t+1}}$. This can be formally written as

$$s_{t+1} \leftarrow s_t + a_t + o_{t+1} \quad .$$

It is assumed that the update is deterministic. Therefore, it must result in one specific dialogue state given the previous state, system dialogue act and the current user dialogue act. Following the example provided in the previous section, an example of the dialogue state update can look like:

Turn 1

Dialogue state:

venue_type	= None
area	= None
price_range	= None
near	= None
food_type	= None
stars	= None

System says: How may I help you?

User says: I am looking for a **restaurant**.
inform(venue_type=”restaurant”)

Turn 2

Updated dialogue state:

venue_type	= **restaurant**

area	= None
price_range	= None
near	= None
food_type	= None
stars	= None

System says: What type of food are you looking for?

User says: I want a **Chinese** restaurant in a **cheap** price range.
inform(venue_type=”restaurant”, food_type=”Chinese”,price_range=”cheap”)

Updated dialogue state:

venue_type	= restaurant
area	= None
price_range	= **cheap**
near	= None
food_type	= **Chinese**
stars	= None

Note that the value “None” denotes that user did not say anything about the relevant slot.

Ontology

The structure of a dialogue state can be defined by some form of ontology. In its simplest form, the ontology can be defined as list of slots and the possible values for the slots.

More complex ontologies can define dependencies between the slots themselves. Some times, not all variable are applicable at all states. For example, the food_type variable is applicable only when the requested venue is a restaurant. Similarly, the variable stars is applicable only when the requested venue is a hotel. Therefore the slot variables can be structured in into some hierarchies or better dependency networks.

In addition, the ontology can define what the user or the system can talk or ask about.

The ontology in UFAL dialogue systems defines:

- all slots in the domain
- the dependencies between slots
- what slots can a user request
- what slots can a user confirm
- what slots can a system
- what information user

the dialogue state can store information about what the user requested or wants to confirm, what the system already informed about or what the system confirmed.

Applicability of the dialogue state variables