Speech Communication 83 (2016) 81–93

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Spoken dialog systems based on online generated stochastic finite-state transducers

Lluís-F. Hurtado*, Joaquin Planells, Encarna Segarra, Emilio Sanchis

Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, E-46022 València, Spain

ARTICLE INFO

Article history:
Received 2 April 2015
Revised 28 July 2016
Accepted 28 July 2016
Available online 1 August 2016

Keywords:
Spoken dialog systems
Dialog management
Statistical models
Stochastic finite-state transducers
User simulation
Coverage problems

ABSTRACT

In this paper, we present an approach for the development of spoken dialog systems based on the statistical modelization of the dialog manager. This work focuses on three points: the modelization of the dialog manager using Stochastic Finite-State Transducers, an unsupervised way to generate training corpora, and a mechanism to address the problem of coverage that is based on the online generation of synthetic dialogs. Our proposal has been developed and applied to a sport facilities booking task at the university. We present experimentation evaluating the system behavior on a set of dialogs that was acquired using the Wizard of Oz technique as well as experimentation with real users. The experimentation shows that the method proposed to increase the coverage of the Dialog System was useful to find new valid paths in the model to achieve the user goals, providing good results with real users.

1. Introduction

1.1. Background literature

A dialog system can be viewed as a human-machine interface that recognizes and understands speech input and generates a spoken answer in successive turns in order to achieve a goal, such as obtaining information or carrying out an action. The development of spoken dialog systems is one of the main objectives of spoken language technology research. Voice-driven applications such as in-car navigation systems or telephone information services are common examples of spoken dialog systems. Most dialog systems are oriented to restricted domain tasks, mixed initiative, and telephone access; however, several new applications have appeared in portable devices such as mobile phones or tablets.

Different modules are necessary to be able to carry out the final goal of a spoken dialog system: a Speech Recognition module converts the audio signal into words; an Understanding module converts these words into a semantic representation; a Dialog Manager decides the next system action in order to fulfill the user's needs; an Answer Generator converts the action of the system into one or more sentences; and a Text-to-Speech Synthesizer converts the system sentences into audio. Each module has its own characteristics and the selection of the most convenient model for it varies depending on certain factors: the goal of each module, the possibility of manually defining the behavior of the module, or the capability of automatically obtaining models from training samples. The use of statistical techniques for the development of the different modules that compose the dialog system has been of growing interest over the last years. These methodologies have traditionally been applied within the fields of Automatic Speech Recognition and Natural Language Understanding (Esteve et al., 2003; Hahn et al., 2010; He and Young, 2003; Levin and Pieraccini, 1995; Minker et al., 1999; Raymond and Riccardi, 2007; Segarra, 2002; Tür and Mori, 2011).

The Dialog Manager is in charge of selecting the action that the dialog system must perform at each turn. This is usually done by taking into account the last user turn and the history of the dialog. Thus, a Dialog Manager can be seen as a function that maps the user turn to an action. Even though there are models for dialog management in the literature that are manually designed using hand-written rules, over the last years, approaches that use statistical models to represent the behavior of the Dialog Manager have been providing compelling results. Statistical models can be trained from real dialogs, modeling the variability of user behaviors.

In the literature, statistical models have been successfully used to select the system action for each user turn. These include Multilayer Perceptrons (Griol et al., 2008; Hurtado et al., 2006), Maximum a Posteriori classifiers (Hurtado et al., 2005), Example-Based modelization (Lee et al., 2007), Bayesian networks (Martinez et al., 2009; Meng et al., 2003; Paek and Horvitz, 2000), Finite State

* Corresponding author. Fax: +34963877359.
E-mail addresses: lhurtado@dsic.upv.es (Lluís-F. Hurtado), xplanells@dsic.upv.es (J. Planells), esegarra@dsic.upv.es (E. Segarra), esanchis@dsic.upv.es (E. Sanchis).

http://dx.doi.org/10.1016/j.specom.2016.07.011
0167-6393/

Transducers (Hori et al., 2009; Hurtado et al., 2010), and Partially Observable Markov Decision Prossess (Williams and Young, 2007). These approaches are usually based on modeling the different processes probabilistically and learning the parameters of the different statistical models from a dialog corpus. In Griol et al. (2014), a technique for automatic acquiring dialog corpora in which the simulated dialogs are automatically generated, is applied to develop a Dialog Manager based on Multi-layer Perceptron classifiers for four different tasks.

In a previous work, we presented an approach to dialog management based on the use of Multi-layer Perceptrons (Griol et al., 2008). That approach have in common with the one presented in the current work the use of a data structure -dialog register- that contains all the information provided by the user throughout the dialog without considering the order in which the information is provided. The main difference between the two approaches is that in the current proposal the next action to be taken by the system at a given point of the dialog is determined not only by the previous history of the dialog but also by a look-ahead mechanism that estimates the quality of the possible finalizations of the dialog from this point.

One of the most commonly used approaches for dialog modeling is based on the use of Partially Observable Markov Decision Processes (POMDPs) (Jurčíček et al., 2012; Williams and Young, 2007). The algorithms of parameter estimation used in POMDPs are mainly based on Reinforcement Learning techniques (Sutton and Barto, 1998).

A first approach that uses Reinforcement Learning techniques for the estimation of the Dialog Manager consists of modeling human-computer interaction as an optimization problem using Markov Decision Processes (MDPs) (Levin and Pieraccini, 1997; Levin et al., 2000; Singh et al., 1999).

Partially Observable MDPs (POMDPs), which are an extension of the MDPs, outperform MDP-based dialog strategies. In POMDPs, the dialog state is not known with certainty (as opposed to MDPs); therefore, the model needs to have a representation for the distribution of the dialog states (belief states). The main drawback of these approaches is the large state space required by practical spoken dialog systems, whose representation is intractable if represented directly (Young et al., 2007). Therefore, those approaches are limited to small-scale problems. In the last few years, many studies have been conducted to implement practical spoken dialog systems based on POMDPs. An approach that scales the POMDP framework by the definition of two state spaces is presented in Young et al. (2010). Another approach based on the use of hierarchical optimization is presented in Cuayáhuitl et al. (2007). Also, an approach that uses a Bayesian update of the dialog states has been presented in Thomson and Young (2010). Besides the computational complexity, the POMDP models need a large number of dialogs to learn good policies. This has been addressed in part by using Gaussian Processes and directly learning from interactions with users (Gašić et al., 2011).

The approaches based on POMDPs maintain active in the belief state all possible states with their attributes values. Differently from them, in the current proposal we modelize the uncertainty by means of a confidence score for each attribute-value pair –belief at attribute level. Thus, we have active only one state at each point of the dialog allowing our approach to deal with realistic state space sizes.

Statistical models have in common their need to have enough training samples to estimate parameters. Since it is very difficult and time consuming to obtain labeled dialog corpora, even with the Wizard of Oz technique, many works have been developed to automatically generate training dialogs (Ai and Litman, 2011; Georgila et al., 2005; Hurtado et al., 2007; Keizer et al., 2010; Schatzmann et al., 2006). This is usually done by developing a user simulation module that interacts with a preliminary system that can be manually defined or obtained by a bootstrapping process.

An interesting initiative to evaluate this kind of systems was the Dialog State Tracking Challenge (Williams et al., 2013). In it, differently from previous Spoken Dialog Evaluation challenges (Black et al., 2010), the interaction progress is measured in terms of finding the correct dialog state that describes the result of the interaction until a certain time *t*. The dialog state tracker takes as input all of the observable elements up to time *t* in a dialog, including all of the results from the automatic speech recognition and spoken language understanding components, and external knowledge sources such as databases and models of past dialogs. It also takes as input a set of possible dialog state hypotheses, where a hypothesis is an assignment of values to slots in the system. The tracker outputs a probability distribution over the set of hypotheses. This is adequate to evaluate heterogeneous Dialog Managers, in particular, those based on POMDPs that works considering multiple state hypotheses.

The dialog model proposed in this article is based on the transduction concept and on the use of Stochastic Finite-State Transducers (*SFST*) (Casacuberta and Vidal, 2004). This approach is based on the assumption that the entire dialog history can be condensed into a finite representation (a dialog state). Based on this state and on the user utterance, the system outputs an answer and moves to another state. Preliminary versions of this work have been presented (Hurtado et al., 2010; Planells et al., 2012).

1.2. Our approach to dialog management

In this approach, given a state of the model and a user turn, a system action is selected and a transition to a new state is performed. Therefore, dialog management is based on the modelization of the sequences of system action and user turn pairs. Then, a dialog describes a path in the transducer model from its initial state to a final one.

Since the space of all combinations of possible sequences of system action and user turn pairs is very large, we establish a partition in the space of sequences of pairs. We define a data structure, called Dialog Register (*DR*), that contains a summary of the information (concepts and attribute values) that is provided by the user throughout the previous history of the dialog. Since the same information can be provided in different order, different sequences of pairs can lead to the same *DR*. We represent the history of the dialog throughout the corresponding *DR*. This representation makes the estimation of a statistical model from the training data manageable. This reduction in the space of all the histories of the dialogs had previously been introduced in another dialog system proposed in our laboratory (Griol et al., 2008). In recent years, within the framework of the POMDPs for dialog management, there have been some approaches to state compression to reduce the search space and make the dialog management problem tractable. Among these works we can highlight (Crook and Lemon, 2011; Cuayáhuitl et al., 2011), where the state compression is done after the state space definition. In our approach the definition of the *DR* implicitly includes a reduction of state space.

In many dialog applications that interact with an Information System, especially where the system is limited to provide the information required by the user, the Dialog Manager has sufficient information supplied by the *DR* to decide the next system action. However, in the case of tasks where the system can update the data of the Information System as a result of interaction with the user (as is the case of EDECAN-SPORTS task (Hurtado et al., 2012)), the dialog manager needs to have additional information to take decisions about its next action. These kinds of more complex dialog systems include an Application Manager, which is a module that communicates the Dialog Manager with the Information