Mon.P1d Spoken Language Understanding and Dialog I

13:30–15:30 Exhibition Hall

Mon.P1d.01 13:30–15:30 Portability of Semantic Annotations for Fast Development of Dialogue Corpora

Bassam Jabaian, Fabrice Lefèvre, Laurent Besacier

Generalization of spoken dialogue systems increases the need for fast development of spoken language understanding modules for semantic tagging of speaker's turns. Statistical methods are performing well for this task but require large corpora to be trained. Collecting such corpora is expensive in time and human expertise. In this paper we propose a semi automatic annotation process for fast production of dialogue corpora. The approach consists in automatically pre-annotating the corpus and then manually correct the annotation. To perform the pre-annotation we propose to port an existing corpus and to adapt it to the new data. The French MEDIA dialogue corpus is used as a starting point to produce two new corpora: one for a new language (Italian) and another for a new domain (theatre ticket reservation). We show that the automatic pre-annotation leads to a significant gain in productivity compared to a fully manual annotation and thus allow to derive new adaptation data which can be used to further improve the systems.

Mon.P1d.02 13:30–15:30 Optimization of Dialog Strategies using Automatic Dialog Simulation and Statistical Dialog Management Techniques

Zoraida Callejas, Ramon Lopez-Cozar

In this paper, we present a technique for learning optimal dialog management strategies. An automatic dialog generation technique, including a simulation of the communication channel, has been developed to acquire the required data, train dialog models, and explore new dialog strategies in order to learn the optimal one. A set of quantitative and qualitative measures has been defined to evaluate the quality of the strategies learned. We provide empirical evidence of the benefits of our proposal through its application to explore the space of possible dialog strategies for the EDECAN spoken dialog system.

Mon.P1d.03 13:30–15:30 Preference-learning based Inverse Reinforcement Learning for Dialog Control

Hiroaki Sugiyama, Toyomi Meguro, Yasuhiro Minami

Dialog control have recently been realized with reinforcement learning, which requires a reward function that is difficult to set appropriately. To set the appropriate reward function automatically, we propose preference-learning based inverse reinforcement learning (PIRL) that estimates a reward function from dialog sequences and their pairwise-preferences calculated with annotated ratings to the sequences. Inverse reinforcement learning finds a reward function, with which a system generates the same sequences as the training ones. This indicates that current IRL supposes that the sequences are equally appropriate for a given task; thus, it cannot utilize the ratings. In contrast, our PIRL can utilize pairwise preferences of the ratings to estimate the reward function. We examine the advantages of PIRL through comparisons between competitive algorithms that have been widely used to realize the dialog control. Our experiments show that our PIRL outperforms the other algorithms and has a potential to be an evaluation simulator of dialog control.

Mon.P1d.04 13:30–15:30 A Data-driven Approach to Understanding Spoken Route Directions in Human-Robot Dialogue

Raveesh Meena, Gabriel Skantze, Joakim Gustafson

Autonomous robots should be able to find directions and navigate their way to a destination in unknown urban environments by seeking route directions from passersby, like humans do. In this paper, we present a data-driven approach for automatic interpretation of spoken route directions into a route graph that may be useful for robot navigation. The results indicate that our approach is robust in handling speech recognitions error and it is indeed possible to get people to freely describe route directions.

Mon.P1d.05 13:30–15:30 Detecting System-directed Utterances Using Dialogue-level Features

Kazunori Komatani, Akira Hirano, Mikio Nakano

We have developed a method to determine whether a user utterance is directed at the system or not. A spoken dialogue system should not respond to audio inputs that are not directed at it (i.e., a user's mutter), and it therefore needs to detect such inputs to avoid unsuitable responses. We classify the two cases by logistic regression based on a feature set including utterance timing, utterance length, and dialogue status. We conducted experiments using 5395 user utterances for both transcription and automatic speech recognition results. Results showed that the classification accuracy improved by 11.0 and 4.1 points, respectively. We also discuss which features are effective in the classification.

Mon.P1d.06 13:30–15:30 An Online Generated Transducer to Increase Dialog Manager Coverage

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

This paper presents a new approach for dynamically increasing the coverage of a Statistical Dialog Manager. A Stochastic Finite-State Transducer for dialog management is estimated using a dialog simulator. This corpus-based model can cover most typical user behavior; however, sometimes unexpected situations may arise. Whenever these situations occur, the Dialog Manager model has no information to determine the next action. To deal with this problem, an Online Dialog Simulator is used in order to obtain synthetic dialogs for re-estimating the model and allowing it the dialog to continue. This approach has been evaluated with real users in a sport facilities booking task.

An Online Generated Transducer to Increase Dialog Manager Coverage

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

Departament de Sistemes Informàtics i Computació. Universitat Politècnica de València
Camí de Vera s/n, 46022, València, Spain.
{xplanells, lhurtado, esanchis, esegarra}@dsic.upv.es

Abstract

This paper presents a new approach for dynamically increasing the coverage of a Statistical Dialog Manager. A Stochastic Finite-State Transducer for dialog management is estimated using a dialog simulator. This corpus-based model can cover most typical user behavior; however, sometimes unexpected situations may arise. Whenever these situations occur, the Dialog Manager model has no information to determine the next action. To deal with this problem, an Online Dialog Simulator is used in order to obtain synthetic dialogs for re-estimating the model and allowing the Dialog Manager to continue the dialog. This approach has been evaluated with real users in a sport facilities booking task.

Index Terms: spoken dialog systems, user simulation, dialog management, coverage problems

1. Introduction

Spoken dialog systems is one of the main fields of the spoken language technologies research. Voice-driven applications are used more and more in our everyday life. Applications such as in-car navigation systems or telephone information services are common examples of spoken dialog systems. A dialog system can be seen as a human-machine interface that recognizes and understands the 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. Most of the dialog systems are oriented to restricted domain tasks, mixed initiative, and telephone access although several new applications have appeared in portable devices like mobile phones or tablets.

The application of statistical methodologies to model the behavior of the dialog manager has provided compelling results in more recent years [1, 2, 3, 4]. The authors' approach is based on Stochastic Finite-State Transducers (SFST) [5]. That is, given a system state and a user turn, a system action is selected and a transition to a new state is done. Therefore, dialog management is based on the modelization of the sequences of system actions and user dialog turn pairs. Then, a dialog describes a path in the transducer model from its initial state to a final one.

In order to estimate the SFST parameters, a large number of labeled dialogs is required. A simulator is used due to the high cost of acquiring dialogs with real users. A process in which dialogs are automatically simulated has been developed [5]. This procedure consists of randomly generating sequences of user dialog-act and system dialog-act pairs. We consider a dialog act, not a label that represents the general intention of the turn, but a frame structure with concepts and pairs (attribute, value). Once a turn sequence has been generated, a set of correctness criteria is applied to classify the dialog as valid or not. All the valid simulated dialogs are used to learn the SFST parameters.

In general, statistical methodologies for dialog management have a reasonable performance in a laboratory environment, but they can present some problems when they are applied to more realistic environments. They have to deal with the lack of robustness against unexpected user utterances, or with relevant recognition or understanding errors. Although a large number of labeled training samples of dialogs are automatically supplied for the estimation of the dialog manager parameters, coverage problems can arise in dialog management when using a transduction model. In a dialog with a real user the dialog manager can get to a situation (a state of the transducer and a user turn) that was never seen in the training samples, then, the dialog manager has no information on what action should be performed next. As a first approach, an ad-hoc heuristic based on the dialog history was used to work around this problem [5].

In this work, the authors propose the use of Online Dialog Simulation to solve coverage problems. In a dialog with a real user, every time an unseen situation occurs, a simulator is used to obtain a set of valid dialogs that share the first turns with the current dialog. These dialogs are used to estimate the transition probabilities for the current state in the dialog model, and then the most likely action is performed. We call a SFST with on-the-fly state estimation an Online Generated Transducer (OGT).

Some experiments applying this technique to the development of a dialog system for a sport facilities booking task were performed. These experiments confirm that this approach has a reasonable behavior with real users and can be used for domain-specific dialog systems.

2. SFST for Dialog Management

In some dialog systems, the dialog manager takes its decisions based only on the information provided by the user in the previous turns and its own model; however, there are dialog systems dealing with more complex tasks. This is the case of the sport facilities booking task dealt in this work, where the dialog manager not only provides information but also modifies the application data (i.e. after booking or canceling a court). For this task, the dialog manager generates the following system answer taking into account not only the information provided by the user, but also the information generated by the module that controls the sport facilities booking application (that we call *Application Manager*, AM).

Within the spoken dialog system framework, the Dialog Manager is the module dedicated to select, from all the possibilities, the best system answer after each user utterance. Recently, an approach to dialog management based on the use of Stochastic Finite-State Transducers was presented. More details abouts this adaptation of SFST models for dialog management can be found in [5].

This approach uses a data structure called *Dialog State* (DS) that contains all the information required by the Dialog Manager for the system answer selection. More precisely, the DS contains: all the information provided by the user throughout the dialog which is stored in the so-called *Dialog Register* (DR), the last request to the AM, and a representation of the last database query result: none (0), a single row (1) or more than one row (2+).

This approach considers a dialog (a sequence of user and system dialog acts) as a path in a SFST from the initial state to one of the final states. A sub-dialog can be seen as a path from a dialog state to another (not necessarily initial or final state).

3. Offline Dialog Simulation

Due the high cost of acquiring dialogs for this task with real users, in order to get a large number of labeled dialogs, a dialog simulator was used. The simulator uses a separate task-independent module for each interaction component: a user simulator, a communication channel simulator, and a dialog manager simulator.

The user simulator has no prior information about user behavior. At the beginning of the dialog, this module chooses an arbitrary goal. During the simulation it randomly selects a user dialog act each turn. This is accomplished selecting some random concepts and a subset of attributes for each one. Random confidence values in $[0, 1]$ are attached to each element. Since the user turn is represented as a frame, there is no need to choose real values for the attribute, so a default value *CORRECT* is used as attribute value.

Each user turn is then modified using the communication channel simulator. For each confidence value in the turn, this module randomly generates a value in the range $[0, 1]$. If the new value is greater than the user's, an error is introduced in that concept or attribute. For attribute values, the *CORRECT* value is replaced by *ERROR* in order to be able later to know that there has been a misunderstanding. For attributes or concept types, another type is selected at random. This ensures that low confidence elements are more likely to be confused than higher confidence ones.

The dialog manager simulator chooses an action at random from the set of actions defined for the task. If the selected action requires a database query, a random value in $\{0, 1, 2+\}$ is used to represent the answer cardinality of the query result. No channel simulation is used for manager turns.

This approach to automatic dialog simulation only requires the semantic definition of the task and a set of criteria to determine the correctness of the generated turn sequence. Given a complete dialog, the correctness criteria set is a function that checks if the dialog is coherent and the user goal is met. This function performs some tests over the turn sequence and the user goal and rejects every dialog that fails at least one test. Some of these tests are: dialog length, checking that no database query has used any *ERROR* values, no system actions unrelated to the user goal, etc.

Thousands of random dialogs are needed to get a single correct dialog, but the procedure is fast enough to get a large corpus in a few minutes and the procedure is easy to parallelize. The term *generated or simulated* dialog refers to a coherent or valid dialog. Those are the only dialogs that are considered for estimating the model. Invalid dialogs are discarded during simulation and not considered anymore.

4. Online Dialog Simulation

Given enough dialogs, every DS should have been visited enough times to give a reliable parameter estimation. In the limit, the model can manage every possible dialog situation as the user simulator is not constrained by any prior knowledge about the task and decide what action to perform at any point. However, even with a dialog simulator, an exponential number of samples are needed in order to get full coverage of the Dialog State space.

The coverage problem arises in dialog management because, when interacting with a real user, errors in the recognition or understanding module can lead to a situation (a state of the transducer and a user turn) that the model has never seen in the training samples and therefore it has no information on what action should be performed next.

The dialog simulator procedure presented here allows us to generate a set of labeled samples given some fixed turns or a sub-dialog. That is, the simulator provides a corpus of coherent dialogs where all dialogs have the

same turns at chosen positions. If the first n turns of the dialog are fixed, a set of correct endings for that situation can be generated.

Using this idea, the authors added an Online Dialog Simulator to the SFST model and obtain the online transducer. This simulator is triggered whenever a dialog with real users gets to a state where the model has no information on what to do next. At this point, the generator is used to create a small corpus of dialogs and then the model transitions are estimated with the new information.

Let $u_0, a_0, u_1, a_1, \ldots, u_i$ be the initial sequence of user turns and system turns of a dialog that reaches a state q_i for which u_i is an unseen situation.

These prefix turns are used to simulate a corpus of k synthetic sub-dialogs that begin from state q_i, end in a final state, and which first user turn is u_i. From this corpus, the dialog manager gets a set of possible actions for q_i: $\{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$, some of them can be repeated, and therefore, more likely. Then, the transducer output function for q_i is estimated from these samples and the most likely action is performed. In the implementation, the next action is selected using the simulated dialogs, but the model parameters are not updated here because the system can not know whether or not that action is correct.

A requirement for this approach is that it needs to be fast enough to accomplish a fluent interaction with the user. It is not realistic to pretend to simulate one thousand dialogs each time the model gets to an unseen situation because a typical user is not going to wait more than a few seconds for a system answer; therefore a small k need to be chosen.

5. Evaluation

5.1. The EDECAN-SPORT task

A sport facilities booking task was defined within the framework of the EDECAN project [6]. This task is called EDECAN-SPORT, and it consists of mixed-initiative dialogs where users can book a court, cancel a booking, search for available facilities, or view their own bookings.

The semantic definition for user turns is a frame structure that includes the different functionalities required for the task: A set of 4 task-dependent concepts representing user intentions (*Booking*, *Cancellation*, *Availability*, *Booked*), and 3 task-independent concepts (*Acceptance*, *Rejection*, *Not-Understood*). Up to 6 attributes can be attached to each concept (*Sport*, *Hour*, *Date*, *Court-Type*, *Court-Number*, *Order-Number*).

Dialog Manager answers are represented using a set of 21 actions. There are actions for opening and closing the dialog, confirming user supplied attributes, asking for more information, or showing information to the user. Each action can have some attributes with the same names as the user frame attributes.

During the corpus acquisition process, a specific Wizard of Oz (WOz) was used to play the role of the natural language understanding module and a second WOz was used to control the dialog manager. As a result of this acquisition process, we obtained not only the dialog corpus but also the dialog acts corresponding to the labeling of the user and system turns.

An initial set of 143 dialogs by 16 different speakers from different origins was acquired for this task. The languages involved in the acquisition were Spanish, Catalan, and Basque. A set of 15 types of scenarios was defined in order to cover all the expected use cases of the task.

5.2. Experiments

Two sets of experiments were defined. The first set was carried out in order to compare the quality of the two versions of the dialog manager: the SFST that was learned from an increasing number of dialogs presented in [5], and the OGT presented in this work. The second set was an evaluation of the OGT approach with real users.

For the first set of experiments, a corpus of $200,000$ dialogs was created using the Offline Dialog Simulator described in Section 3. These dialogs were used to train a SFST model. The number of different DS was $55,645$. A second set of $1,000$ dialogs was simulated and used to train a SFST model for the online generator. We chose a small size for the training set thus, the system had to solve coverage problems in a rate of 33% approximately when tested with real users; the number of different states was $3,378$. The purpose is to show that the model presented here can resolve the coverage problem and can improve the quality of the dialog system. The EDECAN-SPORT corpus described above was used as test set to evaluate the behavior of the two versions of the Dialog Manager.

To evaluate the performances of both managers, the following measures are used:

- Out of model turns. Percentage of dialog turns in the test set for which the DM has no answer.
- Exact turn rate. Percentage of turns in the test set for which the answer of the transducer is equal to the WOz answer during the acquisition.
- Exact dialog rate. Percentage of dialogs in the test set for which in all their states the answer of the transducer is equal to the one produced by the WOz during the real acquisition.
- Completed dialog rate. Percentage of dialogs in the test set accepted by the transducer. That is, the percentage of dialogs in the test set for which there is a path between the initial state and a final state in the transducer.

Table 1 compares the SFST with $1,000$ and $200,000$ dialogs against the SFST with $1,000$ dialogs and the OGT.

Table 1: Evaluation with the EDECAN-SPORT corpus

	SFST		OGT
Training dialogs	1,000	200,000	1,000
States	3,121	55,645	3,378
Out-of-model turns	0.456	0.067	0.329
Exact turn rate	0.343	0.750	0.871
Exact dialog rate	0.007	0.266	0.610
Completed dialog rate	0.084	0.734	1.0

For this experiment, the system simulated 10 dialogs for each *Out-of-model turns* ($k = 10$), this value was the largest possible value of k that allowed real-time answer. It should be noted that although the number of simulated dialogs in the first and third columns are the same $(1,000)$, they are actually two different training sets. Therefore, both the number of *States* and the *Out-of-model turns* are different. The first two columns have been extracted from [5].

As Table 1 shows, the new model improves every measure. Both *Exact turn rate* and *Exact dialog rate* in column three outperform the results in columns one and two. Moreover, the new model is able to deal with every possible situation, so it managed to finalize all the dialogs in the test set although not every dialog met the user goal.

An evaluation of the OGT with real users using our Spoken Dialog System prototype was also carried out. A corpus of 90 dialogs was acquired from 9 users from the University staff with an average of 7.58 dialog system turns. For each dialog, a scenario was selected and explained to the user. The user's mission was to try to achieve a goal by interacting with the dialog system. Examples of such goals are: booking a court some day of the week or checking the user bookings. Some dialogs had more complex goals with combinations of bookings, cancelations and queries that the user must perform in predefined order to achieve the goal.

Table 2 shows the results of the evaluation with real users. From 945 system turns, 233 were *Out-of-model turns* (28.9%) and triggered the Online Dialog Simulator. Every action resulting from this online simulation was locally correct or coherent, that is, does not lead the dialog to an unrecoverable incorrect state like booking a wrong court. The goal was achieved by the user in 94.2% of the dialogs. However, 13.3% of them were not successfully closed by the system, and the user had to manually finish the dialog.

Table 2: Evaluation with real users

Number of users	12
Number of dialogs	120
ASR Word Error Rate	5.97
User turns	804
System turns	945
Out-of-model turns	233 (28.9%)
Successful dialogs	113 (94.2%)

The evaluation shows that the model is robust enough for real-life interaction and that possible recognition and understanding errors are corrected during the dialog using confirmations. Better results are obtained using a model trained with $1,000$ dialogs with the OGT than a SFST estimated with $200,000$ dialogs.

6. Conclusions

This work presents a new approach to increase dynamically the coverage of a Statistical Dialog Manager. When a coverage problem arises, the Online Dialog Simulation process allows the system to continue the dialog. The evaluation compared the performance of this approach with a Stochastic Finite-State Transducer; the results shows a significant improvement in the overall dialog system performance. This new approach also shows a satisfactory behavior when tested with real users.

7. Acknowledgement

This work is partially supported by the Spanish MICINN under contract TIN2011-28169-C05-01, and by the Vic. d'Investigació of the UPV under contract 20100982.

8. References

[1] F. Jurcicek, B. Thomson, S. Keizer, F. Mairesse, M. Gasic, K. Yu, and S. Young, "Natural belief-critic: A reinforcement algorithm for parameter estimation in statistical spoken dialogue systems," in *Proc. of InterSpeech'10*, Makuhari, Japan, 2010, pp. 90–93.

[2] D. Griol, L. F. Hurtado, E. Segarra, and E. Sanchis, "A statistical approach to spoken dialog systems design and evaluation," *Speech Communication*, vol. 50, no. 7-9, pp. 666–682, 2008.

[3] J. Williams and S. Young, "Partially Observable Markov Decision Processes for Spoken Dialog Systems," in *Computer Speech and Language 21(2)*, 2007, pp. 393–422.

[4] O. Lemon, K. Georgila, and J. Henderson, "Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the talk towninfo evaluation," in *Proc. of SLT'06*, 2006.

[5] L.-F. Hurtado, J. Planells, E. Segarra, E. Sanchis, and D. Griol, "A stochastic finite-state transducer approach to spoken dialog management," in *Proc. of InterSpeech'10*, Makuhari, Japan, 2010, pp. 3002–3005.

[6] E. Lleida, E. Segarra, M. I. Torres, and J. Macías-Guarasa, "EDECÁN: sistEma de Diálogo multidominio con adaptación al contExto aCústico y de AplicaciónN," in *IV Jornadas en Tecnologia del Habla*, Zaragoza, Spain, 2006, pp. 291–296.