
MANAGING MULTI-TASK DIALOGS BY MEANS OF A STATISTICAL DIALOG MANAGEMENT TECHNIQUE

PREPRINT

David Griol, Zoraida Callejas, José F. Quesada
Universidad de Granada
Universidad de Sevilla
{dgriol, zoraida}@ugr.es, jquesada@us.es

This is a pre-print version of the chapter: Griol, D., Callejas, Z., Quesada, J.F. (2021). Managing Multi-task Dialogs by Means of a Statistical Dialog Management Technique. In: Marchi, E., Siniscalchi, S.M., Cumani, S., Salerno, V.M., Li, H. (eds) Increasing Naturalness and Flexibility in Spoken Dialogue Interaction. Lecture Notes in Electrical Engineering, vol 714. Springer, Singapore. https://doi.org/10.1007/978-981-15-9323-9_6 (https://link.springer.com/chapter/10.1007/978-981-15-9323-9_6)

This preprint follows Springer Self-archiving policy for non-open access books and chapters (<https://www.springer.com/gp/open-access/publication-policies/self-archiving-policy>): “authors may deposit a portion of the pre-submission version of their manuscript (preprint) in a recognised preprint server (...). This portion of the pre-submission manuscript (preprint) may be deposited and made publicly available at any point.”

ABSTRACT

One of the most demanding tasks when developing a dialog system consists of deciding the next system response considering the user's actions and the dialog history, which is the fundamental responsibility related to dialog management. A statistical dialog management technique is proposed in this work to reduce the effort and time required to design the dialog manager. This technique allows not only an easy adaptation to new domains, but also to deal with the different subtasks for which the dialog system has been designed. The practical application of the proposed technique to develop a dialog system for a travel-planning domain shows that the use of task-specific dialog models increases the quality and number of successful interactions with the system in comparison with developing a single dialog model for the complete domain.

1 Introduction

Spoken conversational interfaces [1] are becoming a strong alternative to traditional graphical interfaces which might not be appropriate for all users and/or applications. These systems can be defined as computer programs that receive speech as input and generate synthesized speech as output, engaging the user in a dialog that aims to be similar to that between humans. Usually, these systems carry out five main tasks: Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialog Management (DM), Natural Language Generation (NLG), and Text-To-Speech Synthesis (TTS).

Learning statistical approaches to model these tasks has been of growing interest during the last decade [2]. Models of this kind have been widely used for speech recognition and also for language understanding. Even though in the literature there are models for dialog managers that are manually designed, over the last few years, approaches using statistical models to represent the behavior of the dialog manager have also been developed [3, 4, 5].

However, statistical dialog modeling and parameterization are dependent on expert knowledge, and the success of these approaches is dependent on the quality and coverage of the models and data used for training [6]. To address these important problems, it is important to develop statistical dialog management methodologies able to infer the dialog structure, which implies detecting if users have changed the topic or dialog task, and to deal with unseen situations (i.e., situations that may occur during the dialog and that were not considered during training).

Research on data-driven approaches to dialog structure modeling is relatively new and focuses mainly on recognizing a structure of a dialog as it progresses [7]. Dialog segmentation can be then defined as the process of dividing up a dialog by one of several related criteria (speaker's intention, topic flow, coherence structure, cohesive devices, etc.), identifying boundaries where the discourse changes taken into account such as specific criteria. This detection is usually based on combining different kinds of features, such as semantic similarities, inter-sentence similarities, entity repetition, word frequency, prosodic and acoustic characteristics.

In this paper we propose a practical implementation of a recently developed statistical approach for the development of dialog managers [4], which is mainly based on the use of a classification process for the estimation of a statistical model from the sequences of the system and user actions obtained from a set of training data. The paper is specially focused on the use of specialized dialog models learned for each dialog domain and dialog subtask, instead of learning a generic dialog model for the complete dialog system. To do this, the training data is divided into different subsets, each covering a specific dialog objective or subtask. These specific dialog models are selected by the dialog manager once the objective of the dialog has been detected, using the generic dialog model until this condition has been fulfilled.

We have applied the proposed methodology to develop two versions of a dialog system providing travel-planning information in Spanish. The first one uses a generic dialog model and the second one combines specific classifiers learned for each dialog objective. An in-depth comparative assessment of the developed systems has been completed by means of recruited users. The results of the evaluation show that the specific dialog models allow a better selection of the next system responses, thus increasing the number and quality of successful interactions with the system.

The rest of the paper is organized as follows. Section 2 describes existing approaches for the development of dialog managers, paying special attention to statistical approaches. Section 3 describes our proposal for developing statistical dialog managers with specific dialog models. Section 4 shows the practical implementation of our proposal to develop the two systems for the customer

support service. In Section 5 we discuss the evaluation results obtained by comparing the two developed systems. Finally, in Section 6 we present the conclusions and outline guidelines for future work.

2 State of the art

As described in the previous section, machine learning approaches to dialog management try to reduce the effort and time required by hand-craft dialog management strategies and, at the same time, to facilitate both to develop new dialog managers and to adapt them to deal with new domains [8].

The most widespread methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem using Markov Decision Processes (MDP) and reinforcement methods [9]. The main drawback of this approach is that the large state space of practical spoken dialog systems, makes its direct re-presentation intractable [10]. Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty [11]. This enables the dialog manager to avoid and recover from recognition errors by sharing and shifting probability mass between multiple hypotheses of the current dialog state.

Other interesting approaches for statistical dialog management are based on modeling the system by means of Hidden Markov Models [12], stochastic Finite-State Transducers [13], or using Bayesian Networks [14]. Also [15] proposed a different hybrid approach to dialog modeling in which n-best recognition hypotheses are weighted using a mixture of expert knowledge and data-driven measures, using an agenda and an example-based machine translation approach respectively.

In the literature, there are different methodologies for the application of statistical methodologies for discourse segmentation and the construction of dialog models including task/subtask information. Unsupervised clustering and segmentation techniques are used in [16] to identify concepts and subtasks in task-oriented dialogs.

Diverse machine-learning methodologies have been recently proposed for dialog state tracking (DST) [17, 18], a similar task whose objective is to use the system outputs, user’s utterances, dialog context and other external information sources to track what has happened in a dialog. Bayesian dynamic networks are used in generative methods to model a dialog [5]. The main drawback of these methods are that additional dependencies and structures must be learned to consider potentially useful features of the dialog history. The parameters for discriminative methods are directly tuned using machine learning and labeled dialog corpus [19]. Recurrent Neural Networks (RNNs) have been recently proposed as to deal with the high dimensional continuous input features involved in sequential models [20].

3 Our proposed methodology for dialog management

This section summarizes the proposed dialog management technique and the practical implementation proposed in this paper by means of specific classifiers adapted to each dialog subtask.

3.1 Proposed statistical methodology

As described in the introduction section, to develop the *Dialog Manager*, we propose the use of specialized dialog models dealing with each one of the subdomains or subtasks for which the dialog system has been designed.

Our proposed technique for statistical dialog modeling represents dialogs as a sequence of pairs (A_i, U_i) , where A_i is the output of the system (the system response or turn) at time i , and U_i is the semantic representation of the user turn (the result of the understanding process of the user input) at time i ; both expressed in terms of dialog acts [21]. This way, each dialog is represented by:

$$(A_1, U_1), \dots, (A_i, U_i), \dots, (A_n, U_n)$$

where A_1 is the greeting turn of the system (e.g. Welcome to the system. How can I help you?), and U_n is the last user turn (i.e., semantic representation of the last user utterance provided by the natural language understanding component in terms of dialog acts).

The lexical, syntactic and semantic information associated with the speaker u ’s i th turn (U_i) is denoted as c_i^u . This information is usually represented by:

- the words uttered;

- part of speech tags, also called word classes or lexical categories. Common linguistic categories include noun, adjective, and verb, among others;
- predicate-argument structures, used by SLU modules in various contexts to represent relations within a sentence structure.
- named entities: sequences of words that refer to a unique identifier. This identifier may be a proper name (e.g., organization, person or location names), a time identifier (e.g., dates, time expressions or durations), or quantities and numerical expressions (e.g., monetary values, phone numbers).

Our model is based on the one proposed in [22]. In this model, each system response is defined in terms of the subtask to which it contributes and the system dialog act to be performed.

The term A_i^a denotes the system dialog act (i.e., system action) in the i th turn, and ST_i^a denotes the subtask label to which the i th turn contributes. The interpretation process is modeled in two stages. In the first stage, the system dialog act is determined from the information about the user’s turn and the previous dialog context, which is modeled by means of the k previous utterances. This process is shown in Eq.(1).

$$A_i^a =_{A^a \in \mathcal{A}} P(A^a | ST_{i-1}^{i-k}, A_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (1)$$

where c_i^u represents the lexical, syntactic, and semantic information (e.g., words, part of speech tags, predicate-argument structures, and named entities) associated with speaker u ’s i th turn; ST_{i-1}^{i-k} represents the dialog subtask tags for utterances $i - 1$ to $i - k$; and A_{i-1}^{i-k} represents the system dialog act tags for utterances $i - 1$ to $i - k$.

In a second stage, the dialog subtask is determined from the lexical information, the dialog act computed according to Eq.(1), and the dialog context, as shown in Eq.(2).

$$ST_i^a =_{s^a \in \mathcal{S}} P(s^a | ST_{i-1}^{i-k}, A_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (2)$$

The prediction of the dialog subtask (ST_i^a) by means of Eq.(2) is carried out by a specific component in the architecture, which we have called the *Task-Dependent Feature Extractor*. This module is connected with the State of the Dialog Management component, which updates the current state of the dialog according to the semantic information provided by the *Natural Language Understanding* module after each user utterance. This information is provided to the Task-Dependent Feature Extractor for the prediction of the dialog subtask. According to this prediction, the *Task-Dependent Feature Extractor* selects the specialized dialog agent that will be used by the dialog manager in the following turn of the dialog. Then, the selected specialized agent employs the corresponding statistical dialog model to select the next action of the dialog system.

In our proposal, we consider static and dynamic features to estimate the conditional distributions shown in Eq.(1) and Eq.(2). Dynamic features include the dialog act and the task/subtask. Static features include the words in each utterance, the dialog acts in each utterance, and predicate-arguments in each utterance. All pieces of information are computed from corpora using n-grams, that is, computing the frequency of the combination of the n previous words, dialog acts, or predicate-arguments in the user turn.

The conditional distributions shown in Eq.(1) and Eq.(2) can be estimated by means of the general technique of choosing the maximum entropy (MaxEnt) distribution that properly estimates the average of each feature in the training data [22]. This can be written as a Gibbs distribution parameterized with weights λ as Eq.(3) shows, where V is the size of the label set, X denotes the distribution of dialog acts or subtasks (DA_i^u or ST_i^u) and ϕ denotes the vector of the described static and dynamic features used for the user turns from $i - 1 \dots i - k$.

$$P(X = st_i | \phi) = \frac{e^{\lambda_{st_i} \cdot \phi}}{\sum_{st=1}^V e^{\lambda_{st_i} \cdot \phi}} \quad (3)$$

Such calculation outperforms other state of the art approaches [22], as it increases the speed of training and makes possible to deal with large data sets. Each of the classes can be encoded as a bit vector such that, in the vector corresponding to each class, the i th bit is one and all other bits are zero. Then, V -one-versus-other binary classifiers are used as Eq.(4) shows.

$$P(y|\phi) = 1 - P(\bar{y}|\phi) = \frac{e^{\lambda_y \cdot \phi}}{e^{\lambda_y \cdot \phi} + e^{\lambda_{\bar{y}} \cdot \phi}} = \frac{1}{1 + e^{-\lambda'_{\bar{y}} \cdot \phi}} \quad (4)$$

where $\lambda_{\bar{y}}$ is the parameter vector for the anti-label \bar{y} and $\lambda'_{\bar{y}} = \lambda_y - \lambda_{\bar{y}}$.

Figure 1 shows the described scheme for the practical implementation of the proposed dialog management technique and its interaction with the rest of the modules in the dialog system.

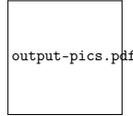


Figure 1: Scheme of the complete architecture for the development of multitask dialog systems

4 Practical application

We have applied our proposal to develop and evaluate an adaptive system for a travel-planning domain. The system provides context-aware information in natural language in Spanish about approaches to a city, flight schedules, weather forecast, car rental, hotel booking, sightseeing and places of interest for tourists, entertainment guide and theater listings, and movie showtimes. Different PostgreSQL databases are used to store this information and automatically update the data that is included in the application. In addition, several functionalities are related to dynamic information (e.g., weather forecast, flight schedules) directly obtained from webpages and web services providing this information. This way, our system provides a speech access to facilitate this travel-planning information, which is adapted to each user taking into account context information.

Semantic knowledge is modeled in the system using the classical frame representation of the meaning of the utterance. We defined eight concepts to represent the different queries that the user can perform (*City-Approaches*, *Flight-Schedules*, *Weather-Forecast*, *Car-Rental*, and *Hotel-Booking*, *Sightseeing*, *Movie-Showtimes*, and *Theater-Listings*). Three task-independent concepts have also been defined for the task (*Affirmation*, *Negation*, and *Not-Understood*). A total of 101 system actions (DAs) were defined taking into account the information that the system provides, asks or confirms.

Using the *City-Approaches* functionality, it is possible to know how to get to a specific city using the different means of transport. If specific means are not provided by the user, then the system provides the complete information available for the required city. Users can optionally provide an origin city to try to obtain detailed information taking into account this origin. Context information taken into account to adapt this information includes user's current position, and preferred means of transport and city.

The *Flight-Schedules* functionality provides flight information considering the user's requirements. Users can provide the origin and destination cities, ticket class, departure and/or arrival dates, and departure and/or arrival hours. Using the *Weather-Forecast* it is possible to obtain the forecast for the required city and dates (for a maximum of 5 days from the current date). For both functionalities, this information is dynamically extracted from external webpages. Context information taken into account includes user's current location, preferred dates and/or hours, and preferred ticket class.

The *Car-Rental* functionality provides this information taking into account users' requisites including the city, pick-up and drop-off date, car type, name of the company, driver age, and office. The provided information is dynamically extracted from different webpages. The *Hotel-Booking* functionality provides hotels which fulfill the user's requirements (city, name, category, check-in and check-out dates, number of rooms, and number of people).

The *Sightseeing* functionality provides information about places of interest for a specific city, which is directly extracted from the webpage designed for the application. This information is mainly based on users recommendations that have been incorporated in this webpage. The *Theater-Listings* and *Movie-Showtimes* respectively provides information about theater performances and movie showtimes that takes into account the users requirements. These requirements can include the city, name of the theater/cinema, name of the show/movie, category, date, and hour. This information is also considered to adapt both functionalities and then provide context-aware information.

A set of 25 scenarios were manually defined to cover the different queries to perform to the system including different user requirements and profiles. Basic scenarios defined only one objective for the

dialog; it means, the user must obtain information about only one type of the possible queries to the system (e.g., to obtain flight schedules from an origin city to a destination for a specific date). More complex scenarios included more than one objective for the dialog (e.g., to obtain information about how to get to a specific city, car rental and hotel booking information).

Two versions of the system have been developed. The first one (*Dialog System 1*) uses a generic dialog model for the task, which employs a single classifier to select the next system response. The second one (*Dialog System 2*) employs 25 specific dialog models, each one of them focused on the achievement of the objective(s) defined for a specific scenario.

5 Results and discussion

We have completed a comparative evaluation of the two practical dialog systems developed for the task. A total of 150 dialogs were recorded from interactions of 25 users employing the two dialog systems. An objective and subjective evaluation were carried out.

The following measures were defined in the objective evaluation to compare the dialogs acquired with the dialog systems: i) Dialog success rate; ii) Dialog length: average number of turns per dialog, number of turns of the shortest dialog, number of turns of the longest dialog, and number of turns of the most observed dialog; iii) Different dialogs: percentage of different dialogs with respect to the total number of dialogs, and number of repetitions of the most observed dialog; iv) Turn length: average number of actions per turn; v) Participant activity: number of turns in the most observed, shortest and longest dialogs; v) Confirmation rate, computed as the ratio between the number of explicit confirmation turns and the total number of turns in the dialog; and vi) Error correction rate, computed as the number of errors detected and corrected by the dialog manager divided by the total number of errors.

Table 1 presents the results of the objective evaluation. As can be observed, both dialog systems could interact correctly with the users in most cases for the two systems. However, the *Dialog System 2* obtained a higher success rate, improving the initial results by a 6% absolute. Using the *Dialog System 2*, the average number of required turns is also reduced from 24.3 to 19.1.

| | <i>Dialog System 1</i> | <i>Dialog System 2</i> |
|----------|------------------------|------------------------|
| M_1 | 89.0% | 95.0% |
| M_2 | 24.3 | 19.1 |
| M_3 | 84.6% | 88.7% |
| M_4 | 4 | 3 |
| M_5 | 1.2 | 1.5 |
| M_6 | 12 | 10 |
| M_7 | 9 | 6 |
| M_8 | 15 | 11 |
| M_9 | 38% | 36% |
| M_{10} | 0.89% | 0.94% |

Table 1: Results of the high-level dialog measures. Dialog success rate (M_1), Average number of turns per dialog (M_2), Percentage of different dialogs (M_3), Repetitions of the most observed dialog (M_4), Average number of actions per turn (M_5), Number of user turns of the most observed dialog (M_6), Number of user turns of the shortest dialog (M_7), Number of user turns of the longest dialog (M_8), Confirmation rate (M_9), Error correction rate (M_{10})

It can also be observed that when *Dialog System 2* was used, there was a reduction in the average number of turns and in the number of turns in the longest, shortest and most observed dialogs. These results show that the use of specialized dialog models made it possible to reduce the number of necessary system actions to attain the dialog goals for the different tasks. In addition, the results show a higher variability in the dialogs generated with *Dialog System 2* as there was a higher percentage of different dialogs and the most observed dialog was less repeated. There was also a slight increment in the mean values of the turn length for the dialogs collected with *Dialog System 2* due to the better selection of the system actions in the improved strategy.

The confirmation and error correction rates were also improved by using *Dialog System 2* as it required less data from the user, thus reducing the number of errors in the automatic speech recognition process. A problem occurred when the user input was misrecognized but it had high confidence score, in

which case it was forwarded to the dialog manager. However, as the success rate shows, this problem did not have a remarkable impact on the performance of the dialog systems.

Additionally, we grouped all user and system actions into three categories: “goal directed” (actions to provide or request information), “grounding” (confirmations and negations), and “other”. Table 2 shows a comparison between these categories. As can be observed, the dialogs provided by the *Dialog System 2* have a better quality, as the proportion of goal-directed actions is higher than the values obtained for the *Dialog System 1*.

| | <i>Dialog System 1</i> | <i>Dialog System 2</i> |
|-----------------------|------------------------|------------------------|
| Goal-directed actions | 68.21% | 74.35% |
| Grounding actions | 30.76% | 24.76% |
| Rest of actions | 1.03% | 0.89% |

Table 2: Proportions of dialog spent on-goal directed actions, ground actions and other possible actions

We also asked the users to complete a questionnaire to assess their subjective opinion about the system performance. The questionnaire had six questions: i) Q1: *How well did the system understand you?*; ii) Q2: *How well did you understand the system messages?*; iii) Q3: *Was it easy for you to get the requested information?*; iv) Q4: *Was the interaction with the system quick enough?*; v) Q5: *If there were system errors, was it easy for you to correct them?*; vi) Q6: *In general, are you satisfied with the performance of the system?* The possible answers for each one of the questions were the same: *Never/Not at all, Seldom/In some measure, Sometimes/Acceptably, Usually/Well, and Always/Very Well*. All the answers were assigned a numeric value between one and five (in the same order as they appear in the questionnaire). Table 3 shows the average results of the subjective evaluation using the described questionnaire.

| | <i>Dialog System 1</i> | <i>Dialog System 2</i> |
|----|------------------------|------------------------|
| Q1 | 4.7 | 4.8 |
| Q2 | 4.3 | 4.4 |
| Q3 | 4.2 | 4.7 |
| Q4 | 4.2 | 4.6 |
| Q5 | 4.1 | 4.3 |
| Q6 | 4.4 | 4.7 |

Table 3: Results of the subjective evaluation with recruited users (1 = lowest, 5 = highest)

It can be observed that using either *Dialog System 1* or *Dialog System 2* the users perceived that the system understood them correctly. Moreover, they expressed a similar opinion regarding the easiness for correcting system errors. However, users said that it was easier to obtain the information specified for the different objectives using *Dialog System 2*, and that the interaction with the system was more adequate with this dialog manager. Finally, the users were more satisfied with the system employing *Dialog System 2*.

We have completed the evaluation with an additional assessment using a dialog simulation technique [23], which allows to develop a user simulator to automatically interact with a conversational system. The user simulator emulates the user intention, that is, the simulator provides concepts and attributes that represent the intention of the user utterance. Therefore, the user simulator carries out the functions of the ASR and NLU modules. An error simulator module is also integrated to perform error generation and the addition of confidence measures [24].

A total of 3000 dialog were simulated for the two dialog systems developed. The dialogs were only considered successful if they fulfilled the complete list of objectives that had been previously defined for the simulation. Table 4 shows the values obtained for the task success/efficiency measures considered. As it can be observed, the percentage of successfully simulated dialogs increases when the dialog task segmentation module is included. Our analysis also shows that not only the dialogs of the systems including the dialog task segmentation (DTS) module achieve their goals more frequently, but also their average completion time is shorter. The number of different simulated dialogs that are obtained is also increased when this module is included.

Table 4: Results of the high-level dialog features defined for the comparative assessment of the simulated dialogs

| | <i>Dialog System 1</i> | <i>Dialog System 2</i> |
|----------------------------|------------------------|------------------------|
| Success. simulated dialogs | 83.7% | 87.8% |
| Different dialogs | 76.7% | 82.9% |
| Repetit. most seen dialog | 11 | 7 |
| Avg. number of user turns | 6.2 | 5.1 |

6 Conclusions and future work

In this paper, we have described a statistical technique for dialog management in dialog systems. Our proposal is based on dealing with each one of the dialog subtasks or dialog objectives by means of a specific dialog model specialized in each one of them. This model, which considers the previous history of the dialog, is used for the selection of each specialized dialog agent according to the predicted dialog subtask, and the decision of the next system action. Although the construction and parameterization of the dialog model depends on expert knowledge of the task, by means of our proposal, we facilitate to develop dialog systems that have a more robust behavior, better portability, and are easier to be extended or adapted to different user profiles or tasks.

The results of the evaluation of our proposal for a travel-planning dialog system show that the number of successful dialogs is increased in comparison with using a generic dialog agent learned for the complete task. Also, the dialogs acquired using the specific dialog agents are statistically shorter and present a better quality in the selection of the system responses. For future work, we want to consider the incorporation of additional information regarding the user, such as specific user profiles related to their emotional states and adapted to the each application domain.

7 Acknowledgements

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 823907 (MENHIR project:<https://menhir-project.eu>) and the CAVIAR project (MINECO, TEC2017-84593-C2-1-R, AEI/FEDER, UE).

References

- [1] M.F. McTear, Z. Callejas, and D. Griol. *The Conversational Interface*. Springer, 2016.
- [2] S. Young. Talking to Machines. *Royal Academy of Engineering Ingenia*, 54:40–46, 2013.
- [3] P. Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech and Language*, 34(1):232–255, 2015.
- [4] D. Griol, J.A. Iglesias, A. Ledezma, and A. Sanchis. A Two-Stage Combining Classifier Model for the Development of Adaptive Dialog Systems. *Int. Journal of Neural Systems*, 26(1), 2016.
- [5] J. Williams and S. Young. Partially Observable Markov Decision Processes for Spoken Dialog Systems. In *Computer Speech and Language 21(2)*, pages 393–422, 2007.
- [6] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review*, 21(2):97–126, 2006.
- [7] Y. Zhou, Q. Hu, J. Liu, and Y. Jia. Combining heterogeneous deep neural networks with conditional random fields for chinese dialogue act recognition. *Neurocomputing*, 168:408–417, 2015.
- [8] M. Frampton and O. Lemon. Recent research advances in reinforcement learning in spoken dialogue systems. *The Knowledge Engineering Review*, 24(4):375–408, 2009.
- [9] E. Levin and R. Pieraccini. A stochastic model of human-machine interaction for learning dialog strategies. In *Proc. of the European Conference on Speech Communications and Technology (Eurospeech’97)*, pages 1883–1896, Rhodes, Greece, 1997.
- [10] Steve Young, Jost Schatzmann, Karl Weilhammer, and Hui Ye. The Hidden Information State Approach to Dialogue Management. In *Proc. of the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 149–152, Honolulu, Hawaii, USA, 2007.

- [11] N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, pages 93–100, Hong Kong, China, 2000.
- [12] H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'05)*, pages 290–295, San Juan, Puerto Rico, 2005.
- [13] J. Planells, L.F. Hurtado, E. Sanchis, and E. Segarra. An Online Generated Transducer to Increase Dialog Manager Coverage. In *Proc. of the International Conference on Spoken Language Processing (Interspeech'2012)*, Portland, USA, 2012.
- [14] Heleng H. Meng, Carmen Wai, and Roberto Pieraccini. The Use of Belief Networks for Mixed-Initiative Dialog Modeling. *IEEE Transactions on Speech and Audio Processing*, 11(6):757–773, 2003.
- [15] Cheongjae Lee, Sangkeun Jung, Kyungduk Kim, and Gary Geunbae Lee. Hybrid approach to robust dialog management using agenda and dialog examples. *Computer Speech and Language*, 24(4):609–631, 2010.
- [16] A. Chotimongkol. *Learning the structure of task-oriented conversations from the corpus of in-domain dialogs*. PhD thesis, CMU, Pittsburgh (USA), 2008.
- [17] J.D. Williams, A. Raux, and M. Henderson. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33, 2016.
- [18] J. Pérez and F. Liu. Dialog state tracking, a machine reading approach using Memory Network. In *Proc. of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 305–314, Valencia, Spain, 2017.
- [19] A. Metallinou, D. Bohus, and J. Williams. Discriminative state tracking for spoken dialog systems. In *Proc. of ACL'13*, pages 466–475, Sofia, Bulgaria, 2013.
- [20] Pei-Hao Su, Nikola Mrksic, Iñigo Casanueva, and Ivan Vulic. Deep learning for conversational AI. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2018)*, pages 27–32, New Orleans, Louisiana, USA, 2018.
- [21] D. Griol, Z. Callejas, R. López-Cózar, and G. Riccardi. A domain-independent statistical methodology for dialog management in spoken dialog systems. *Computer, Speech and Language*, 28(3):743–768, 2014.
- [22] S. Bangalore, G. DiFabrizio, and A. Stent. Learning the Structure of Task-Driven Human-Human Dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259, 2008.
- [23] D. Griol, J. Carbó, and J.M. Molina. A statistical simulation technique to develop and evaluate conversational agents. *AI Communications*, 26(4):355–371, 2013.
- [24] J. Schatzmann, B. Thomson, and S. Young. Error Simulation for Training Statistical Dialogue Systems. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'07)*, pages 273–282, Kyoto, Japan, 2007.