

This is a postprint version of the following published document:

Griol, D., Iglesias, J. A., Ledezma, A. and Sanchis, A.
(2016). A Two-stage Combining Classifier Model for
the Development of Adaptive Dialog Systems.
International Journal of Neural Systems, 26(1)

DOI: [10.1142/S0129065716500027](https://doi.org/10.1142/S0129065716500027)

© World Scientific Publishing Company



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

A TWO-STAGE COMBINING CLASSIFIER MODEL FOR THE DEVELOPMENT OF ADAPTIVE DIALOG SYSTEMS

DAVID GRIOL, JOSÉ ANTONIO IGLESIAS, AGAPITO LEDEZMA, ARACELI SANCHIS

Control Learning and Systems Optimization Group

Computer Science Department

Carlos III University of Madrid

Avda. de la Universidad, 30

28911 Leganés, Spain

This paper proposes a statistical framework to develop user-adapted spoken dialog systems. The proposed framework integrates two main models. The first model is used to predict the user's intention during the dialog. The second model uses this prediction and the history of dialog up to the current moment to predict the next system response. This prediction is performed with an ensemble-based classifier trained for each of the tasks considered, so that a better selection of the next system can be attained weighting the outputs of these specialized classifiers. The codification of the information and the definition of data structures to store the data supplied by the user throughout the dialog makes the estimation of the models from the training data and practical domains manageable. We describe our proposal and its application and detailed evaluation in a practical spoken dialog system.

Keywords: Spoken Dialog Systems; Dialog Management; User Models; Classifier Systems; Artificial Neural Networks; Clustering; Spoken Human-Machine Interaction.

1. Introduction

Spoken dialog systems are computer programs that receive speech as input and generate as output synthesized speech, engaging the user in a dialog that aims to be similar to that between humans.^{59,78,67,81} Thus, these interfaces make technologies more usable, as they ease interaction with devices such as smartphones and tablets,⁶⁹ allow integration in different environments,³⁴ facilitate the interaction with different systems⁸⁶ and make technologies more accessible, especially for disabled people.^{107,20}

The performance of spoken language dialog systems has improved over time, extending the initial application domains to complex information retrieval and question answering applications,⁷¹ surveys applications,¹⁰⁰ e-commerce systems,¹⁰⁵ recommendations systems,⁸⁵ e-learning and tutoring systems,⁵⁴ in-car systems,³⁸ spoken dialog within vehicles,⁷³ remote control of devices and robots in smart envi-

ronments,⁷² Ambient Assisted Living systems,⁸ or embodied dialog systems and companions.³⁹

The described systems are usually designed ad-hoc for their specific domain using rule-based dialog models and standards in which developers must specify the steps to be followed by the system. This way, the adaptation of the hand-crafted designed systems to consider specific users requirements or deal with new tasks is a time-consuming process that implies a considerable effort, with the ever-increasing problem of dialog complexity.^{92,80}

In addition, although much work emphasizes the importance of taking into account user's models not only to solve the tasks presented to the dialog system by the user, but also to enhance the system performance in the communication task, this information is not usually considered when designing the dialog model for the system.^{95,50} For this reason, in most dialog applications, the dialog specification is the same for all cases: users typically have no control over the

content or presentation of the service provided.

Incorporating intelligence into a spoken language based communication system requires, among other things, careful user modeling in conjunction with an effective dialog management. With the aim of creating dynamic and adapted dialogs, the application of statistical approaches to user modeling and dialog management makes it possible to consider a wider space of dialog strategies in comparison to engineered rules.^{119, 29}

The selection of a specific system action depends on multiple factors, such as the output of the speech recognizer (e.g., measures that define the reliability of the recognized information), the dialog interaction and previous dialog history (e.g., the number of repairs carried out so far), the application domain (e.g., guidelines for customer service), knowledge about the users, and the responses and status of external back-ends, devices, and data repositories. Given that the actions of the system directly impact users, the design of an appropriate dialog management strategy is at the core of dialog system engineering.

As an attempt to reduce the time and effort required for system implementation and carry out rapid system prototyping, statistical approaches for dialog modeling and management are gaining increasing interest.^{65, 24} These approaches enable automatic learning of dialog strategies, thus avoiding the time-consuming process that hand-crafted dialog design involves. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. Although the construction and parameterization of these models depend on expert knowledge about the task to be carried out by the dialog system, the final objective is to develop systems that are more robust for real-world conditions, and that are easier to adapt to different users and tasks.^{70, 113, 119}

In this paper we describe a framework to develop user-adapted spoken dialog systems. Our proposal is based on the definition of a statistical methodology for user modeling that estimates the user intention during the dialog. The term *user intention* expresses the information that the user has to convey to the system to achieve their goals, such as extracting some particular information from the system. It is a very useful and compact representation of human-computer interaction that specifies the next steps to be carried out by the user as a counterpart in the

human-machine conversation.

This prediction, carried out for each user turn in the dialog, makes it possible to adapt the system dynamically to the user's needs. To do this, a statistical dialog model based on neural networks is generated taking into account the predicted user's intention and the history of the dialog up to the current moment. The next system response is selected by means of this model. The codification of the information and the definition of a data structure which takes into account the data supplied by the user throughout the dialog makes the estimation of the dialog model from the training data and practical domains manageable.

The current paper presents the following important contributions. One of the main ideas of our proposal is that the dialog system and the user model interact simultaneously, not offline as in our previous works,^{29, 30} so that the user model is employed in real time by the dialog manager to decide the best answer dynamically and not in a posteriori laboratory evaluations. In addition, the technology used to build the proposed user model does not replicate that of the dialog manager. This avoids the dialogs generated to be biased.

The novel user model incorporates numerous information sources to decide the next user action, and integrate and orchestrate these heterogeneous sources and use them significantly to make decisions. To optimize these computations it is very important to estimate the task underlying the current dialog. Determining the task is one of the main innovations of the paper as it is not only used for the user simulator, but also for the practical implementation of the dialog system. Another important contribution is that such implementation is performed with an ensemble-based classifier trained for each of the tasks considered, so that a better selection of the next system can be attained weighting the outputs of these specialized classifiers.

All these decisions have been carefully designed so that the proposal can be portable across domains and applied to systems with varying complexity. In particular, the current paper shows the application of our proposal to the Let's Go task.^{94, 32} This provides the benefit of showing an application of our scientific proposals in a task designed for a real system that provides a real service to real users. Additionally, Let's Go is a reference within the dialog systems

community, which makes our proposal easily verifiable and comparable to other approaches.

Finally, the detailed evaluation showed in the paper covers the specific assessment of the user model and the overall evaluation of the proposal. For example, it applies clustering techniques to evaluate the user model and verify the realism of the simulated behaviors, and computes the relevancy of the features that define such clusters. Furthermore, the overall evaluation of the proposal does not only employ measures to compare the answers provided by the manager offline, but also incorporates an evaluation with real users.

The remainder of the paper is organized as follows. In Section 2 we describe the motivation of our proposal and review main approaches focused on key aspects related to it, such as user modeling techniques when interacting with dialog systems and the application of statistical methodologies for dialog management. Section 3 presents in detail our proposal to develop adaptive dialog systems. Section 4 describes the application of our proposal in the CMU Let’s Go spoken dialog system, a system that has been used during the last years by the dialog systems community as a common ground for comparison and verifiable assessment of the improvements achieved. Section 5 presents the results of the evaluation of our proposal for this practical system. Finally, Section 6 presents the conclusions and suggests some future work guidelines.

2. Related work

The design and development of a comprehensive adaptive spoken dialog system can be conceptually composed of two interconnected components; the user modeling, and the corresponding adaptation that in our proposal is implemented on the dialog manager.

Research in techniques for user modeling has a long history within the fields of language processing and dialog systems. A thorough literature review on the application of how data mining techniques to user modeling for system personalization can be found in [21, 90]. It is possible to classify the different approaches with regard to the level of abstraction at which they model dialog. This can be at either the acoustic level, the word level or the intention-level. The latter is a particularly useful representation of human-computer interaction.⁹⁰

In recent years, simulation on the intention-level has been most popular.⁹⁰ This approach has been adopted for user simulation by most research groups.^{120,47} Modeling interaction on the intention-level avoids reproducing the enormous variety of human language on the level of speech signals or word sequences.⁹⁰

Georgila et al. propose the use of Hidden Markov Models (HMMs), defining a more detailed description of the states and considering an extended representation of the history of the dialog.²⁶ Dialog is viewed as a sequence of Information States,⁹³ each of which is represented as a large feature vector describing the current state of the dialog, the previous dialog history, and any ongoing actions. Dethlefs and Cuayáhuitl define a HMMs-based dialog simulation technique in which both the user and system behaviors are simulated.¹⁹ Instead of training only a generic HMM model to simulate any type of dialog, a submodel is trained for each one of the objectives.

Jung et al. describe a data-driven user intention simulation method that integrates diverse user discourse knowledge (cooperative, corrective, and self-directing).⁴⁶ User intention is modeled based on logistic regression and the Markov logic framework. Higashinaka et al. also propose incorporating discourse features for a more accurate confidence scoring of intention recognition results in slot-based dialog systems.³⁶ To do this, both acoustic and language model features extracted from the words uttered by the user are considered to estimate the confidence scoring of the intention recognition results.

Seon et al. propose a statistical prediction model of the user’s intentions using morpheme-level features, discourse-level features, and domain-level features as inputs to a statistical model based on the Maximum Entropy Model (MEM).⁹⁶ This model allows integrating information from many heterogeneous sources. Each feature corresponds to a constraint and the model employed is the one with maximum entropy that satisfies the constraints. Winterboer et al. propose a user-model based summarize and refine approach that improves task success, efficiency, and user satisfaction with dialog systems.¹¹⁵

Recently, Wang and Swegles propose a technique that employs knowledge about the user’s activity to disambiguate their spoken inputs.¹⁰⁸ A Reinforcement Learning algorithm is proposed to acquire the knowledge and apply it for disambiguation. The

interpreted user utterance is then transmitted to the dialog manager to select the next system response.

Schatzmann et al. propose a technique for user simulation based on explicit representations of the user goal and the user agenda.⁸⁹ This model formalizes human-machine dialogs at a semantic level as a sequence of states and dialog acts for which the user has a predefined plan (agenda) that may vary during the conversation. The user agenda is a stack-like structure containing the pending user dialog acts that are needed to elicit the information specified in a dialog goal. As the dialog progresses the agenda and goal are dynamically updated and the dialog acts are selected from the top of the agenda to form user acts.

As will be described in Section 3.1, our user intention simulation technique considers specific user interactions by incorporating several knowledge sources, combining statistical and heuristic information to enhance the dialog model. Some of its main advantages are the simple integration with the dialog manager and the possibility to use simulated dialogs for cost-effective development.

Once a user model has been generated, it is required to define how to use it to adapt the dialog system. A comprehensive study of dialog management methodologies and architectures is presented in [4, 109, 58]. Statistical models for dialog management can be trained with corpora of human-computer dialogs with the main objective of explicitly modeling the variance in user behavior that can be difficult to address by means of hand-written rules.⁹⁰ Additionally, if it is necessary to satisfy certain deterministic behaviors, it is possible to extend the strategy learned from the training corpus with handcrafted rules that include expert knowledge or specifications about the task.^{101, 57, 104, 120}

The goal is to build systems that exhibit more robust performance, improved portability, better scalability and easier adaptation to other tasks. However, model construction and parameterization is dependent on expert knowledge, and the success of statistical approaches is dependent on the quality and coverage of the models and data used for training.⁹⁰ Moreover, the training data must be correctly labeled for the learning process. The size of currently available annotated dialog corpora is usually too small to sufficiently explore the vast space of possible dialog states and strategies. Collecting a corpus with real users and annotating it requires considerable time

and effort.

To address these problems, researchers have proposed alternative techniques that facilitate the acquisition and labeling of corpora, such as Wizard of Oz,¹² bootstrapping,²³ active learning,¹¹⁴ automatic dialog act classification and labeling,⁷⁷ and user simulation.⁹⁰

Another relevant problem is how to deal with unseen situations, that is, situations that may occur during the dialog and that were not considered during training. To address this point it is necessary to employ generalizable models in order to obtain appropriate system responses that enable to continue with the dialog in a satisfactory way.

Another difficulty is in the design of a good dialog strategy, which in many cases is far from being trivial. In fact, there is no clear definition of what constitutes a good dialog strategy.⁶⁰ Users are diverse, which makes it difficult to foresee which form of system behavior will lead to quick and successful dialog completion, and speech recognition errors may introduce uncertainty about their intention.

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.⁴⁷ The main drawback of this approach is that the large state space of practical domains makes its direct representation intractable.¹²¹ Partially Observable MDPs (POMDPs) outperform MDP-based dialog strategies since they provide an explicit representation of uncertainty.¹⁷ However, they are also limited to small-scale problems, since the state space would be huge and exact POMDP optimization is again intractable.¹²¹

Other authors have combined conventional dialog managers with a fully-observable Markov decision process,³³ or proposed the use of multiple POMDPs and selecting actions using hand-crafted rules.¹¹¹ The robustness of the POMDP can also be combined with conventional approaches.¹¹² Bayesian reinforcement learning frameworks for learning the POMDP parameters have been also recently proposed.⁶³ Other interesting approaches for statistical dialog management and user modeling are based on Hidden Markov Models,¹⁸ stochastic Finite-State Transducers,⁸² Least-Squares Policy Iteration,⁶² evolving classifiers,^{44, 43} or Bayesian Networks.⁷⁹

The use of evolutionary and neural-fuzzy techniques in enhancing the dialog in human-computer interaction systems is currently a very important approach in Natural Language Processing.^{1,13,88,52} Zuckerman et al. have recently presented a numerical mechanism for the interpretation of spoken referring expressions. Their proposal considers multiple alternatives at different interpretation stages (speech, syntax, semantics, and pragmatics) and combines distance-based functions that represent lexical similarity using two approaches, viz multiplicative and additive.¹²⁵ A framework for audio-assisted dialog detection based on indicator functions and neural networks is described in.⁵⁵ Several types of artificial neural networks, including multilayer perceptrons (MLPs),^{3,40,41,42} voted perceptrons, radial basis function networks,^{2,48,49,27} support vector machines,^{14,124} and particle swarm optimization-based MLPs^{74,116,11,97,123} are tested.

Montero and Araki describes a genetic algorithm (GA)-like transfer approach to automatically generate and evaluate dialog sentences.⁷⁵ The evaluation showed an 80% user understandability of the sentences generated by the system. Yahya et al. presents a model for the automatic recognition of dialog acts.¹¹⁷ The proposed model employs a dynamic Bayesian network for which the dynamic random variables are constituted from sets of lexical cues selected automatically by means of a variable length genetic algorithm. Giacobello et al. have recently proposed a formal methodology based on a genetic algorithm for tuning the parameters of a single-microphone speech enhancement system for hands-free devices.²⁸ The proposal greatly improves the enhanced speech quality and also speeds up development and deployment time.

Chen et al. presents a dynamic gesture recognition method for multimodal systems based on the combination of the fuzzy features of the dynamic gesture track changes and the fuzzy neural network inference system.¹⁵ The results of its evaluation show that the proposed method is more applicable to real gesture-controlled human-computer interactive environments. Several works have also recently proposed the use of this kind of algorithms for enhancing dialog systems with emotion recognition and eye-tracking capabilities.^{31,99}

Recent developments in speech research have shown promising results using deep learning.^{61,37}

Henderson et al. presents a Deep Neural Network (DNN) approach for dialog state tracking.³⁵ The results of its practical application show improvements in using deep networks over networks with fewer hidden layers.

Our methodology for dialog management (Section 3.2) is based on the estimation of a statistical model from the user's intention prediction provided by the user's model and sequences of the system and user dialog acts obtained from a corpus of training data. This way, the next system response is selected by means of a classification process that considers the complete history of the dialog and the user model, which is one of the main advantages regarding the previously described statistical methodologies for dialog management. Another benefit is the inclusion of a data structure that efficiently stores the complete information related to the task provided by the user during the dialog history. The main objective of this structure is to easily encode the complete information related to the task provided by the user during the dialog history, then considering the specific semantics of the task and including this information in the proposed classification process.

Over the last two decades, the machine learning and related communities have conducted numerous studies to improve the performance of a single classifier by combining several classifiers generated from one or more learning algorithms.^{122,51,118,68} An ensemble of classifiers is a set of classifiers whose individual decisions are combined to obtain a system that aims to outperform all of its members.^{7,25} Therefore, classifier ensembles are built in two phases: generation and combination.^{1,98} In the generation phase, the individual components of the ensemble, known as base classifiers, are generated. In the combination phase, the decisions made by the members of the ensemble are combined to obtain one decision.^{16,45}

In our proposed framework we combine a set of classifiers to firstly predict the user's intention (user dialog act and current subtask of the dialog), and then take this information into account to select the next system response. This way, an initial classifier is used for the prediction of the user's intention. The estimated user dialog act is used to update an input register that contains the complete list of features provided by the SLU module through the dialog history until the current moment. The predicted

subtask of the dialog is used to ponder the outputs provided for this input register by a set of classifiers, which are specialized in each one of the subtasks in the dialog. The outputs of these classifiers can be seen as the probabilities of selecting each one of the possible system responses.

The classification function can be defined in several ways. We have evaluated seven different definitions of this function in previous works: a multinomial naive Bayes classifier, an n-gram based classifier, a decision tree classifier, a support vector machine classifier, a classifier based on grammatical inference techniques, Fuzzy-rule-based (FRB) classifiers, and a classifier based on artificial neural networks.^{29,30} The best results were obtained using multilayer perceptrons (MLP).^{5,87}

The proposed techniques for the prediction of the user’s intention and the selection of the next response of the system consists thus of two phases. First, the user and dialog models are trained on a dialog corpus to learn the set of parameters for the corresponding classifiers. A form of supervised learning, i.e. “learning by example” is commonly used for this step.⁹⁰ In the second phase the trained models are used to interact with real users and automatically predict the system actions also considering the output of the user model. After each dialog, the learning system can be also automatically based on the feedback given by the dialog. A Reinforcement-Learning process can be also used by means of the interaction of an initial version of the user and dialog models. This technique allows any number of training dialogs to be generated and it also enables dialog strategies that are not present in the initial corpus of human-computer dialogs to be explored.⁹⁰

3. Proposed framework to develop user-adapted spoken dialog systems

Figure 1 shows the architecture that integrates our proposed framework to generate adaptive spoken dialog systems. To successfully manage the interaction with the users, spoken dialog systems usually 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). The goal of the ASR module is to obtain the sequence of words uttered by a speaker. It is a very complex task, as there can be a great deal of variation in the input

the recognizer must analyze (e.g., linguistics of the utterance, inter and intra speaker variation, interaction context, and the communication channel). The goal of the SLU module is to obtain the semantics from the recognized sentence. This process generally requires morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge.

The DM decides the next action of the system, interpreting the incoming semantic representation of the user input in the context of the dialog. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and understanding errors, retrieves information from data repositories, and decides about the next system’s response. Natural language generation is the process of obtaining sentences in natural language from the non-linguistic, internal representation of information handled by the dialog system. Finally, the TTS module transforms the generated sentences into synthesized speech.

In our proposal, a User Modeling module (Section 3.1) is incorporated to consider the previous dialog interactions to predict the user intention, defined as the current dialog subtask and the next user action, which we represent by one or more dialog acts as described in the previous section (i.e., the same standard representation defined for the output provided by the SLU module in the dialog system).

The Dialog Manager (Section 3.2) takes as input this prediction, the current user utterance, and the sequence of user and system dialog acts until the current moment. Using this information it selects the next system action (next system dialog act) by means of a classification process, in which classifiers adapted to each specific subtask are used.

3.1. User Modeling

Our proposed technique for user modeling simulates the user intention providing the next user dialog act in the same representation defined for the spoken language understanding module. We represent 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.²⁹ This way, each dialog is represented by:

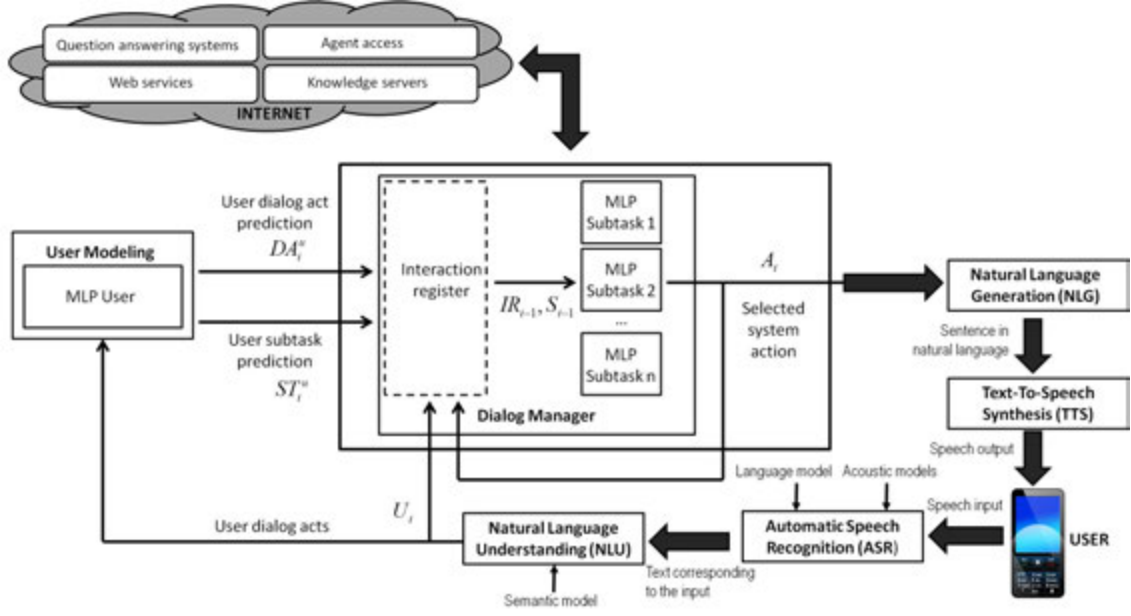


Fig. 1. Architecture to develop adaptive spoken dialog systems

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

where A_1 is the greeting turn of the system (e.g. Welcome to Let's Go 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 SLU module in terms of dialog acts). We refer to a pair (A_i, U_i) as S_i , the state of the dialog sequence at time i .

The lexical, syntactic and semantic information associated to 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. They are usually represented as triples (subject-verb-object).
- 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 dura-

tions), or quantities and numerical expressions (e.g., monetary values, percentages or phone numbers).

Our model is based on the one proposed in [6]. In this model, each user turn is modeled as a user action defined by a subtask to which the turn contributes, the dialog act of the turn, and its named entities. For example, for the Let's Go system, a subtask may be to provide the information necessary to perform a timetable query, the turn may be to provide the origin address, and the dialog act may be *Provide-Street*, being *Queen Avenue* the named entity involved.

For speaker u , DA_i^u denotes the dialog act in the i th turn, and ST_i^u denotes the subtask label to which the i th turn contributes. The interpretation process is modeled in two stages. In the first stage, the dialog act of the clause 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).

$$DA_i^u = \operatorname{argmax}_{d^u \in \mathcal{D}} P(d^u | c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (1)$$

where c_i^u represents the lexical, syntactic, and semantic information (e.g., words, part of speech tags, predicate-argument structures, and named enti-

ties) associated with speaker u 's i th turn; ST_{i-1}^{i-k} represents the dialog subtask tags for utterances $i \cdots 1$ to $i \cdots k$; and DA_{i-1}^{i-k} represents the dialog act tags for utterances $i \cdots 1$ to $i \cdots k$.

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).

$$ST_i^u = \operatorname{argmax}_{s^u \in \mathcal{S}} P(s^u | DA_i^u, c_i^u, ST_{i-1}^{i-k}, DA_{i-1}^{i-k}, c_{i-1}^{i-k}) \quad (2)$$

where the different variables have been explained for the previous equations.

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 (Max-Ent) distribution that properly estimates the average of each feature in the training data.⁶ 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 $i-1 \cdots i-k$.

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

Such calculation outperforms other state of the art approaches,⁶ 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 $\lambda'_{\bar{y}} = \lambda_y - \lambda_{\bar{y}}$.

3.2. Interaction Management

Considering the representation of dialogs as the sequence of pairs (A_i, U_i) described in the previous subsection, at time i , the objective of the dialog manager is to find the best system answer A_i . This selection is a local process for each time i and takes into account the previous history of the dialog:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | S_1, \dots, S_{i-1}) \quad (5)$$

where set \mathcal{A} contains all the possible system answers.

Following Eq.(5), the dialog manager selects the next system response taking into account the sequence of previous pairs (A_i, U_i) . The main problem to resolve this equation is usually the large number of possible sequences of states. To solve the problem, we define a data structure in order to establish a partition in this space, i.e., in the history of the dialog preceding time i). This data structure, which we call *Interaction Register (IR)*, contains the following information: i) sequence of user dialog acts provided by the user throughout the previous history of the dialog (i.e., the output of the SLU module); ii) predicted user dialog act, generated by means of Eq.(1).

After applying these considerations and establishing the equivalence relation in the histories of the dialogs, the selection of the best A_i is given by Eq.(6).

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | IR_{i-1}, S_{i-1}) \quad (6)$$

Each user turn supplies the system with information about the task; i.e., the user asks for a specific concept and/or provides specific values for certain attributes (e.g., to obtain timetables from a specific origin and destination in a bus information system). However, a user turn can also provide other kinds of information, such as task-independent information (e.g., *Affirmation*, *Negation*, and *Not-Understood* dialog acts). This kind of information implies some decisions which are different from simply updating the

IR_{i-1} . Hence, for the selection of the best system response A_i , we take into account the IR from turn 1 to turn $i - 1$, and we explicitly consider the last state S_{i-1} .

For the dialog manager to determine the next system answer, we have assumed that the exact values of the task-dependent attributes are not significant. They are important for accessing data repositories and for constructing the output sentences of the system. However, the only information necessary to predict the next system action is the presence or absence of concepts and attributes (i.e. whether each relevant piece of information has been correctly provided or not). Therefore, the codification we use for this information in the IR is in terms of three values, $\{0, 1, 2\}$, according to the following criteria:

- (0) The concept is unknown or the value of the attribute is not given;
- (1) the concept or attribute is known with a confidence score that is higher than a given threshold. To decide whether the state of a certain value in the IR is 1 or 2, the system employs confidence measures provided by the ASR and SLU modules.¹⁰³
- (2) the concept or attribute has a confidence score that is lower than the given threshold.

We propose to solve Eq.(6) by approximating this equation by a learned function. To do this, every dialog situation is classified taking into account a set of classes \mathcal{C} , in which a class contains all the sequences that provide the same set of system actions (responses). The objective of the dialog manager at each moment is to select a class of this set $c \in \mathcal{C}$, so that the system answer is the one associated with the selected class. The input of the classifiers for each specific dialog subtask holds a codification of the input pair (IR_{i-1}, S_{i-1}) . The representation defined for this pair is as follows:

- Last system response (A_{i-1}): This information is modeled using a variable, which has as many bits as possible system responses (C).

$$\vec{x}_1 = (x_{1_1}, x_{1_2}, x_{1_3}, \dots, x_{1_C}) \in \{0, 1\}^C$$

where C is the number of possible system re-

sponses (i.e., system actions) as it has been previously described.

- Interaction register (IR_{i-1}): As previously stated, the interaction register includes task-dependent user dialog acts and the prediction of the current user dialog act and subtask. Each one of the task-dependent user dialog acts can take the values $\{0, 1, 2\}$ and then be modeled using a variable with three bits. The prediction of the current user dialog act is modeled using a variable, which has as many bits as possible user responses (N).

$$\vec{x}_j = (x_{j_1}, x_{j_2}, x_{j_3}) \in \{0, 1\}^3 \quad j = 2, \dots, N + 1$$

$$\vec{x}_{N+2} = (x_{1_1}, x_{1_2}, x_{1_3}, \dots, x_{1_N}) \in \{0, 1\}^N$$

- Task-independent information (*Affirmation*, *Negation*, and *Not-Understood* dialog acts): These three dialog acts have been coded with the same codification used for the task-dependent information in the IR ; that is, each one of these three dialog acts can take the values $\{0, 1, 2\}$. This information is modeled using three variables with three bits.

$$\vec{x}_j = (x_{j_1}, x_{j_2}, x_{j_3}) \in \{0, 1\}^3 \quad j = N+3, \dots, N+6$$

4. Practical application: the Let's Go spoken dialog system

Let's Go is a spoken dialog system developed by the Carnegie Mellon University to provide bus schedule information in Pittsburgh at hours when the Port Authority phones are not carried out by operators (7pm to 7am on weekdays and 6pm to 7am on weekends). The information provided by the system covers a subset of 5 routes and 559 bus stops.

The system has had many users since it was made available for the general public in 2005, with more than 20,000 calls collected just from March to December of 2005,⁸³ so there is a substantial dataset that can be used to train a dialog model. In addition, this large amount of data from spoken interactions has been acquired with real callers, rather than lab

^a<http://cmusphinx.sourceforge.net>

testers. The system integrates the RavenClaw dialog manager,¹⁰ the Sphinx 2 speech recognition engine^a and a domain-specific voice built with the Festival/Festvox toolkit and deployed on the Cepstral Swift engine^b.

In a study carried out after the first two operative years of the system,⁸⁴ the average daily call traffic for the past year oscillated between 40 and 60. The average length of dialogs was 14 turns. However the distribution of dialog turn lengths was bi-modal, with a first peak at 0 turns (10% of the dialogs) and a second one around 10 turns. Complete dialogs have a 79% dialog success rate. Typical failures detected in the dialogs included system wrongly interrupting its turn (52.0% of calls), system failing to take a turn (47.1% of calls), system failing to yield a turn on user barge-in (43.1% of calls), system taking an extra turn (39.2% of calls), and system wrongly barging in on user (15.7% of calls). At the end of April 2015, the number of calls is typically averaging 40 calls during weekdays and 100 calls during weekends. The current version of the dialog system provides an estimated dialog success of 75.11% with an average number of 12.24 turns, with a relatively large standard deviation of 11.5 turns.⁶⁵

Figure 2 shows an example of a dialog extracted from the Let’s Go corpus.³² Each call to the system starts with a welcome message that prompts the user to make a request. Then, the system waits for the user’s response and grabs concepts such as question type (e.g., *When is the next bus to X?*, *How can I go from X to Y?*) or departure and arrival times and places. To be successful, calls require three or four pieces of information from the user: a departure stop, a destination, a travel time, and, optionally a bus route. Stops can be specified in one of three ways: the nearest intersection to the stop (e.g., Forbes [Avenue] at Murray [Avenue]), a neighborhood (e.g., Oakland), or a landmark or other point of interest (Pittsburgh International Airport, Waterworks Mall). The system explicitly prompts the user to provide the missing information to complete the query. Once the system has the required information to answer the user’s query, it submits a query to the database, presents the results to the user, and prompts for a new query.

<p>System: Welcome to the Pittsburgh bus information line. You can say start over anytime. Where are you going to take your bus from? User: Hi yeah I want to go from McKeesport System: Sure. Where do you want to go? User: Duquesne. System: Did you mean Duquesne? User: Yes System: At what time do you want to take it? Say next bus or a time for example 8.40pm. User: Around 4.15pm System: I’m going to search for buses around 4.15pm. Is that OK? User: Yes. System: On which day? User: Today System: I have found one bus. It is at 4.15pm. There is a bus number 61C from McKeesport. It will arrive in Duquesne at 4.21pm. System: Is there anything else I can do for you? User: No thanks System: Thanks for calling. Good bye!</p>
--

Fig. 2. Example of dialog extracted from the Let’s Go corpus

In 2009, a corpus of 338 dialogs acquired with real users was distributed among the scientific community as a common testbed for the 2010 Spoken Dialog Challenge (SDC) initiative.⁹ The aim of the Challenge was to bring together multiple implementations of the same dialog task and deploy them in uncontrolled real user conditions. Different statistical and rule-based methodologies were employed for the development of these implementations. For instance, the AT&T Let’s Go system¹¹⁰ incorporates the AT&T Statistical Dialog Toolkit (ASDT), which tracks a distribution over many dialog states in real time, and regression-based confidence scores, which are trained on a corpus of in-domain recognitions. The BUDSLETSGO system integrates a universal statistical language model for speech recognition, and a POMDP-based dialog manager that maintains a full probability distribution over all possible user goals and provides mixed-initiative.¹⁰² Selfridge et al. presented an Incremental Interaction Manager approach that also supports the simple integration of incremental speech recognition results with POMDP-based dialog managers.⁹⁴ Hastie et al. presented an implementation based on the Information State Update (ISU) approach and deployed on the VoiceXML industry standard.³²

We have chosen the Let’s Go task to evaluate our proposal for several reasons. Firstly, the corpus available was gathered from a real task in an oper-

^b<http://www.cepstral.com>

ative dialog system that provided its service to real users. This poses a challenge to build realistic user models and find new dialog strategies that are at least as good as the hand-crafted system. Secondly, Let’s Go is a common ground for experimentation and evaluation within the dialog system community, which therefore makes our results directly comparable to the alternatives presented by other authors, and this is why it has been intensively used by researchers in the last years.^{91,9,110,32}

With regard the semantic representation defined for the task, the system uses a set of user dialog acts that has been classified into 16 categories following the criteria described in.¹⁰² A total of 16 categories of user dialog acts were defined. Four of the dialog acts are used to model where the user is leaving from (monument, pair of road names, neighborhood, or stop). The four dialog acts used for modeling the place of arrival are similar. Six dialog acts are used for describing the user’s required time of travel (next bus or specific times). The *meth* node describes whether the user is asking for a bus with some constraints, is finished or wants to restart. The dialog act *disc* models how the user issues “discourse” actions, which relate to only one turn in a dialog.

A total of 36 system dialog acts were defined. These dialog acts can also be classified into 5 groups: *formal* (dialog formalities like “welcome”), *results* (presentation of search results), *queries* (request for values to fill slots), *statusreports* (when the system reports about its status, e.g. “looking up database”), *error* (error messages), and *instructions* (instructions to the user how to speak to the system). Table 1 show different examples of the user and system dialog acts defined to represent semantic information in the Let’s Go task.

Table 1. Different examples of the semantic interpretation defined for Let’s Go

User dialog acts
I want to go from McKeesport <i>PlaceInformation/DeparturePlace/stop_name/covered_place</i>
Around 4.15pm <i>TimeInformation/time_range/time/hour</i>
What about the previous bus? <i>RequestType/BusBeforeThatRequest</i>
System dialog acts
Welcome to the CMU Let’s Go bus information system <i>(Welcome)</i>
Where do you want to go? <i>(Query.departure_place)</i>
Leaving from OAKLAND. Is this correct? <i>(Confirm.departure_place)</i>

The different objectives of the dialogs for the Spoken Dialog Challenge were labeled in the corpus by considering the different places and times for which the users required information (from one to five), users’ requirements about previous and next buses, number of uncovered places, and possible system failures. The different combinations of these parameters in the corpus lead to the definition of 38 different objectives. The dialogs were also divided into 10 subtasks (*welcome*, *ask_for_query*, *ask_for_attribute*, *confirm_query*, *confirm_attribute*, *looking_up_database*, *provide_results*, *provide_instructions*, *query_error*, and *goodbye*).

In order to successfully use neural networks as classifiers, a number of considerations had to be taken into account, such as the network topology, the training algorithm, and the selection of the parameters of the algorithm. Different experiments were conducted using different network topologies of increasing number of weights: a hidden layer with 2 units, two hidden layers of 2 units each, two hidden layers of 4 and 2 units, a hidden layer with 4 units, etc. Several learning algorithms were also tested: the incremental version of the backpropagation algorithm (with and without momentum term) and the quickprop algorithm. The influence of their parameters such as learning rate or momentum term was also studied.

To train and evaluate the neural networks, we used the *April* toolkit.²² We firstly tested the influence of the topology of the MLP, by training different MLPs of increasing number of weights using the standard backpropagation algorithm (with a sigmoid activation function and a learning rate equal to 0.2), and selecting the best topology according to the mean square error (MSE) of the validation data. The minimum MSE value was achieved using an MLP of one hidden layer of 32 units. We followed our experimentation with MLPs of this topology, training MLPs with several algorithms. The best result on the validation data was obtained using the MLP trained with the standard backpropagation algorithm and a value of LR equal to 0.3.

The good operation of the MLP is also fostered in our approach by determining beforehand the value of several input features (e.g., predicted current user dialog act and subtask) by means of the process described in Section 3.1, thus reducing the number of input features for the MLP with respect to state-of-

the-art approaches. The Let’s Go task considers up to 455 different values for the task-dependent user dialog acts. Our codification of the features using only 3 values (0,1,2) makes it possible to reduce the dimensionality of the problem making the dialog manager more efficient and scalable, and suitable for complex application domains.

5. Evaluation

As explained in the previous section, our approach incorporates a module that predicts the user intention, which is then forwarded to the dialog manager as another input feature. This is why we have evaluated our proposal with the Let’s Go dialog system using two experiments: the first one evaluates whether the user modeling module is able to produce realistic user behaviors (Section 5.1), and the second experiment evaluates the appropriateness of the decision of the dialog manager computed considering the user model (Section 5.2).

5.1. Evaluation of the user modeling module

The main objective of the first experiment is to assess the proposed methodology for user modeling. To do this, we applied our proposal to build a user model trained with the initial corpus of 338 dialogs. Then it was used to acquire a second corpus of 1,000 simulated dialogs by means of its interaction with the initial version of the Let’s Go dialog system (DM_1).

Since user models are built to replace human users, a good user model should be able to replicate human user behaviors. Under this assumption, assessing the user model usually consists in a subjective assessment of the realism of the simulated user or measuring the similarity to real user behavior based on more formal criteria.

In order to avoid biases, we selected to use objective criteria and computed a large set of dialog parameters that characterize the real and simulated dialogs. Then, we employed a technique called subspace clustering¹⁰⁶ with which we studied whether the clustering algorithm was able to find significant differences between the simulated and real users, which would indicate that the simulated users behave in a significant different way with respect to the real users. By means of this particular clustering technique it is not necessary to reduce the dimen-

sionality of the space beforehand unlike other state-of-the-art approaches, which would lead to a loss of valuable information.⁵³

Table 2 shows the features computed for the 1,338 dialogs acquired for the Let’s Go task. For each one of the 5 groups of system dialog acts the counts and percentages of each group were calculated as new parameters. For our experiments we have employed the PROCLUS projected clustering algorithm, which detects all the possible clusters in all subspaces. The algorithm builds the clusters taking into account different subsets of the attributes and assigns each dialog to a unique cluster. To do so, we used Opensubspace,⁷⁶ an implementation of the algorithm that can be integrated into the Weka machine learning tool.

Table 2. Features computed for the two kinds of dialogs

<i>numExchanges</i>	Number of exchanges between user and system. An exchange is comprised of a system turn and the successive user turn, where the user turn can be empty (e.g. no-input or when the end of a dialog is reached).
<i>percConfirm</i>	Percentage of confirmation dialog acts.
<i>meanReasks</i> and <i>maxReasks</i>	mean and maximum re-asks.
<i>numNoMatches</i> , <i>percNoMatches</i>	Number and percentage of no-matches
<i>numNoInputs</i> , <i>percNoInputs</i>	Number and percentage of no-inputs
<i>meanAVPs</i>	Mean number of concepts provided in the user utterances. A concept can be a value for a slot, a logical (yes/no), a DTMF (Dual-Tone Multi-Frequency) signal, navigation keywords such as <i>next bus</i> , help requests or a dialog ending action. The total number of concepts is divided by the number of exchanges in the dialog to obtain the mean.
<i>successDial</i>	Task success, determined automatically by checking if the objective of the dialog was reached.
<i>SysActPUserAct</i>	Number of system turns in a dialog divided by the total number of concepts provided by the user.

Table 3 shows the 4 clusters generated. As can be observed, different features have been chosen for each cluster, and thus there are 4-dimensional and bidimensional subspaces. The features selected are mainly related to situations in the dialog which differ from the optimal, such as out-of-vocabulary inputs, silences, number of error messages, percentage of error messages and the percentage of presentation of results.

We have carried out a statistical study of the parameters per cluster computing their maximum, minimum, average and standard deviation (Table 4

shows average values). The study reveals that the dialogs in cluster 1 and cluster 2 (82.21% of the corpus) usually reached their objective, sometimes with the use of re-asks and other techniques in order to solve the possible errors in the input.

Table 3. Results of the subspace clustering for Let’s Go

Cluster (dimensions): relevant features	Dialogs
0 (4D): <i>numNoInputs</i> , <i>percNoInputs</i> , <i>successDial</i> , <i>percResults</i>	59
1 (4D): <i>percNoInputs</i> , <i>meanAVP</i> , <i>numError</i> , <i>percError</i>	861
2 (2D): <i>numNoMatches</i> , <i>numNoInputs</i>	239
3 (2D): <i>percNoInputs</i> , <i>percResults</i>	29

Cluster 1 is comprised of dialogs with no error system dialog acts and with the lowest mean number of concepts provided in each user utterance, while cluster 2 is comprised of dialogs in which the speech understanding phase has an optimal behavior as there are no no-matches and no no-inputs. Cluster 0 and cluster 3 hold the longest dialogs, the case of cluster 3 is peculiar as it holds the dialogs (2.16% of the corpus) with a higher number of no-inputs.

Table 4. Average value of the interaction parameters

Parameters	Clus. 0	Clus. 1	Clus. 2	Clus. 3
<i>numExchanges</i>	18.10	9.76	9.69	30.14
<i>percConfirm</i>	0.62	0.59	0.57	0.66
<i>numNoMatches</i>	3.61	0.40	0.00	10.52
<i>numNoInputs</i>	0.00	0.00	0.00	0.04
<i>meanAVPs</i>	1.03	1.00	0.99	1.04
<i>successDial</i>	1.00	0.97	0.46	1.00
<i>sysActPerUserAct</i>	1.76	2.12	2.33	1.53
<i>meanReasks</i>	1.28	1.09	1.15	1.58
<i>maxReasks</i>	2.78	1.56	1.69	4.24
<i>percNoMatches</i>	0.18	0.03	0.00	0.03
<i>percNoInputs</i>	0.00	0.00	0.00	0.01
Dialog acts	Clus. 0	Clus. 1	Clus. 2	Clus. 3
<i>numFormal</i>	1.19	1.12	1.08	1.03
<i>numResults</i>	1.81	1.43	1.13	1.59
<i>numQueries</i>	13.78	7.99	6.97	22.41
<i>numStatusReports</i>	8.39	5.15	5.22	10.73
<i>numError</i>	1.02	0.00	1.06	3.24
<i>numInstructions</i>	3.39	2.31	2.09	4.17
<i>percFormal</i>	0.08	0.13	0.19	0.05
<i>percResults</i>	0.10	0.15	0.07	0.05
<i>percQueries</i>	0.78	0.82	0.62	0.76
<i>percStatusReports</i>	0.46	0.53	0.42	0.04
<i>percError</i>	0.06	0.00	0.26	0.10
<i>percInstructions</i>	0.20	0.25	0.27	0.15

Clusters 1 and 2 were mainly built based on the simulated data, whereas the real data contributed mostly to clusters 0 and 3. This shows that the features used to define the subspaces for those clusters might indicate a difference in the behavior of the simulated and real dialogs. Concretely the number of no

inputs seems to be relevant, as the presence of such behavior is not common in the simulated dialogs. In any case, as shown in Table 4 the number of no inputs (*numNoInputs* and *percNoInputs* parameters) is very reduced, also these situations represent the most uncommon human user behaviors as cluster 3 only contains 2.16% of the dialogs considered. Thus, we can conclude that the simulated users rendered a realistic behavior which was in most cases not distinguishable from the real users.

5.2. Evaluation of the complete framework

In the second experiment we assessed the appropriateness of the dialog manager decisions taking into account also the result of the user modeling module. A 5-fold cross-validation process was used to carry out the evaluation. The initial corpus of 338 dialogs was randomly split into five subsets of 1,817 samples (20% of the corpus). Our experiment consisted of five trials. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80% of the corpus was used as the training set for the user and dialog models. A validation subset (20%) was extracted from each training set.

Each sample in the corpus code a dialog state and the corresponding system response (i.e., dialog act) annotated in the corpus. As described in the previous section, dialog states are represented as a pair (IR_{i-1}, S_{i-1}) . This pair consists of a total number of 21 features defined for the Let’s Go task: the last system response (A_{i-1}), 17 features corresponding to the Interaction Register IR_{i-1} (predicted current user dialog act and 16 task-dependent user dialog acts), and 3 features corresponding to the task-independent information (*Affirmation*, *Negation*, and *Not-Understood* dialog acts).

From our previous work on statistical dialog management,²⁹ we propose three measures to evaluate the quality of the responses selected by the statistical dialog manager. These measures are calculated by comparing the answer automatically generated by the statistical dialog manager (DM_2) for each sample in the test partition with regard to the reference answer annotated in the corpus (DM_1). This way, the evaluation is carried out turn by turn. Thus, the aim is not to evaluate the complete dialog as a unit, but to assess the appropriateness of the dialog manager response for each sample in the test partition

(i.e., current situations of the dialog). The three measures used for the described evaluation are: i) *Matching*: the percentage of responses provided by the dialog manager that are equal to the reference answer in the corresponding turn of the test corpus; ii) *Coherence*: the percentage of answers provided by the dialog manager that are coherent with the current state of the dialog although they are not necessarily the same that the reference answer; iii) *Error*: the percentage of answers provided by the dialog manager that would cause the failure of the dialog.

The measure *Matching* is automatically calculated, evaluating whether the responses provided by DM_1 and DM_2 are the same. The calculation of the *Coherence* and *Error* measures requires expert annotation of the corpus. Thus, to decide about coherence of system responses, we asked three annotators to answer the following question: “Given the current dialog state: does it make sense that the system generates this response?”. They were also advised about considering user’s adaptation as an important criterion to answer the question. The responses labeled as *Error* correspond to those that have not been considered coherent.

Table 5 shows the results of the proposed evaluation that compared the initial dialog manager for the Let’s Go system (DM_1) and the dialog manager developed using our proposal (DM_2). The values obtained for the matching and coherence measures show that the DM_2 dialog manager deviates from the initial dialog model and provides new valid paths to achieve each one of the required objectives defined in each task. This way, exact matches between DM_1 and DM_2 were reduced while coherence increased, as most of the non-matching responses were coherent and thus acceptable for the task.

Table 5. Evaluation results of the DMs developed for the Let’s Go system

	<i>Matching</i>	<i>Coherence</i>	<i>Error</i>
DM_1	93.28%	94.52%	5.48%
DM_2	81.22%	97.94%	2.37%

A deeper study of the system responses provided by both dialog managers showed that DM_2 by considering the information provided by the user model was able to tackle new situations and generate new coherent answers for the situations already present in the initial corpus. Also it could avoid pre-

viously detected errors anticipating the user’s intention and was better prepared for future user’s actions being able to disambiguate between different alternatives for the user’s dialog acts at each turn. Also, the initial system for the task was designed to be strongly system-directed, with the use of closed system prompts to limit the variability of possible user responses after each one of them. Also, each concept is explicitly confirmed by means of a specific system response for each one of them without considering the confidence scores provided by the ASR module. As described in,^{65,9} this makes the pace of the dialog somewhat slow and, even without recognition errors, expert users might find it frustrating to have to provide and confirm one concept at a time.

The proposed codification of the information in the Interaction Register described in Subsection 3.2 allows considering the annotated reliability of each one of the information pieces provided by the user for a better selection of the next system action. The detected decrease in the average number of turns is also associated to the selection of system responses that are different to the one annotated in the corpus (*Matching* measure), but were considered coherent with the current state of the dialog (*Coherence* measure) and allows to achieve the objective(s) of the dialog in a less number of turns. Moreover, the codification developed to represent the state of the dialog and the good operation of the MLP classifiers make it possible for the number of responses that cause the failure of the system to be only 2.37% for the DM_2 dialog manager, instead of the initial 5.48% in DM_1 .

With respect to the dialog style features, we measured the balance between different types of system dialog acts using DM_1 and DM_2 . The results, showed in Table 6, indicate that using DM_2 there was an increment in the number of system turns that actually provide information to the user, which is consistent with the fact that the task completion rate is higher using our dialog manager.

In addition, we grouped all user and system dialog acts into “goal-directed” (actions to provide or request information) and “grounding” actions (dialog formalities, unrecognized actions, confirmations, and negations). The results in Table 7 show that the dialogs acquired with DM_2 are better as the proportion of goal-directed actions increases for this system. The increase in the number of goal-directed actions

using our proposal is also related to the described deviation of the initial dialog strategy.

Table 6. Percentages of system dialog acts using DM_1 and DM_2

	DM_1	DM_2
S_Confirm	39.16	35.08
S_Request	20.08	18.34
S_Inform	40.39	46.37
S_Other	0.37	0.21

With regard a literature comparison between the proposed methodology and the different studies carried out using the Let’s Go dialog system, as far as we are concerned, there are not previous studies that have developed user models for the task. The initial version of the system provided complete dialogs with a 79% success rate with an average length of a dialog of 14 turns and a minimum number of 6 necessary confirmations.⁸⁴ This produced an average percentage of grounding actions of 38.42 and a value of 42.86 for the S_Confirm measure. The version of the system presented in⁶⁵ provides a 77.64% success rate with an average number of turns of 11.47. With regard a version of the system developed by means of the DUDE development,³² the 62% of calls reached the stage of presenting results to the user. Of these calls, 61% gave fully correct information to the users, and 74% were correct with respect to the route information. The results of the evaluation of the different versions of the system that participated at the 2010 Spoken Dialog Challenge are compiled in.¹⁰² The four systems described in this paper respectively provide success rates of 64.8 ± 5.0 , 37.7 ± 6.2 , 89.3 ± 3.6 , and 74.7 ± 4.8 .

Table 7. Percentages of goal directed and grounding actions using DM_1 and DM_2

	DM_1	DM_2
Goal-directed actions	73.16	79.12
Grounding actions	26.84	20.88

5.3. Evaluation with real users

To assess the benefits of our proposal, we have finally developed two systems for the Let’s Go task: the baseline and the user-adapted systems. The baseline system does not carry out any adaptation to the user, while the user-adapted system incorporates our

proposal as described in Figure 1. The main objective of this evaluation is to assess our proposal with real users and on-line conditions.

In order to do so, 30 recruited users participated in the evaluation, aged 21 to 69 (mean 37.2), 69% male. A total of 120 dialogs was recorded from the interactions of the recruited users, 15 users employed the user-adapted system and 15 users employed the baseline version of the system. The users were provided with a brochure describing the scenarios that they were asked to complete and main functionalities of the system. A total of 38 scenarios was defined to consider the definition of different dialog objectives for the Let’s Go task described in the previous section. Thus, each scenario specified a set of objectives that had to be fulfilled by the user at the end of the dialog and they were designed to include and combine the complete set of functionalities previously described for the system. An objective (Section 5.3.1) and subjective (Section 5.3.2) evaluations were carried out.

5.3.1. Objective evaluation

We considered the following high-level measures for the objective evaluation: 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 8 presents the results of the objective evaluation. As can be observed, both systems could interact correctly with the users in most cases. However, the user-adapted system obtained a higher success rate, improving the initial results by a value of 5% absolute. Using the user-adapted system, the average number of required turns is also reduced from 12.1 to 9.3.

These results show that improving the dialog strategy 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 the user-adapted system 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 the user-adapted system due to the better selection of the system actions in the improved strategy.

The confirmation and error correction rates were also improved by using the user-adapted system as it required less data from the user, thus reducing the number of ASR errors. 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 system performance.

Table 8. High-level dialog measures obtained for the user-adapted and baseline systems. 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), Confirmation rate (M_7), Error correction rate (M_8)

	Baseline system	User-adapted system
M_1	89.0%	94.0%
M_2	12.1	9.3
M_3	77.8%	87.2%
M_4	6	3
M_5	1.2	1.5
M_6	5	4
M_7	39%	36%
M_8	0.88%	0.93%

5.3.2. Subjective evaluation

One of the ultimate goals of designing and building spoken dialog systems is the optimization of the caller experience. This way, 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 9 shows the average results of the subjective evaluation using the described questionnaire.

Table 9. Results of the subjective evaluation with recruited users (1 = lowest, 5 = highest)

Baseline system	4.5	4.5	4.1	4.0	3.6	4.3
User-adapted system	4.6	4.6	4.7	4.6	4.0	4.7

It can be observed that using either the user-adapted or the baseline system 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 the user-adapted system, and that the interaction with the system was more adequate with this dialog manager. Finally, the users were more satisfied with the system employing the user-adapted system.

6. Conclusions and future work

In this paper, we have contributed a framework that can be used to develop adaptive spoken dialog systems. Our proposal is based on the definition of a statistical methodology for user modeling that anticipates the next user turn during the dialog and makes it possible to adapt the system dynamically to the specific subtasks and user’s needs. To do this, a statistical dialog model based on neural networks selects the next system response taking into account the prediction of the user’s intention and the history of the dialog up to the current dialog state.

Our methodology for dialog management is based on the estimation of a statistical dialog model from the sequences of the system and user dialog acts and the prediction of the user’s intention (pre-

dicted next user dialog act). The complete history of the dialog is considered to determine the next system answer. The codification of the information and the definition of a data structure which takes into account the data supplied by the user throughout the dialog makes possible to isolate task-dependent knowledge and apply our proposal to real practical domains.

We have shown that our domain-independent approach is scalable and can help to reduce the dimensionality of complex slot-filling domains with a high number of input parameters. Besides, it can also be used in more open-ended situations in which the user responses are less predictable and the system must take into account additional sources of information. These information sources can be considered transparently in our approach as an additional feature for the selection of the next system action. The proposed classification process to select the next system response allows to generate generalizable dialog models, which can deal with unseen situations, that is, situations that may occur during the dialog and that were not considered during training.

Our proposal allows to build user-adapted spoken dialog systems that exhibit more robust performance, improved portability, better scalability and easier adaptation to other tasks. However, model construction and parameterization is dependent on expert knowledge, and the success of statistical approaches is dependent on the quality and coverage of the models and data used for training. Moreover, the training data must be correctly labeled for the learning process. The size of currently available annotated dialog corpora is usually too small to sufficiently explore the vast space of possible dialog states and strategies. Collecting a corpus with real users and annotating it requires considerable time and effort. To address these problems, researchers have proposed alternative techniques that facilitate the acquisition and labeling of corpora (e.g., Wizard of Oz,⁵⁶ bootstrapping,²³ active learning,⁶⁴ and user simulation⁶⁶).

We have provided a complete implementation of our framework for the Let's Go dialog system, a dialog system that has been widely used in the scientific community for dialog evaluation. With regard to the assessment of the proposed user modeling technique, we have shown that the user model resembles the real user behaviors in the majority of the dialogs consid-

ered, and thus can be used as a reliable input to the dialog manager. With respect to the assessment of the compute proposed framework integrating the dialog manager and the user modeling module, the results show that the number of coherent responses provided by the statistical dialog manager increases with respect to the baseline, while the number of responses that lead to dialog failure decreases. The dialog manager also improves the confirmation and error correction rates for the different tasks.

For future work we plan to apply the proposed technique to other tasks in order to see whether it can be used for comparison between several user models and dialog management techniques. We also intend to extend the evaluation of the system considering user profiles and satisfaction measures that complement the proposed adaptation and the statistical measures employed.

Acknowledgments

We would like to thank Maxine Eskenazi, Alan Black, Lori Levin, Rita Singh, Antoine Raux and Brian Langner from the Let's Go Lab at Carnegie Mellon University, Pittsburgh, for providing the Let's Go Sample Corpus. We also thank Zoraida Callejas for her help and valuable comments.

References

1. H. Adeli and S. Hung, *Machine Learning - Neural Networks, Genetic Algorithms, and Fuzzy Systems* (John Wiley and Sons, 1995).
2. H. Adeli and A. Karim, Fuzzy-Wavelet RBFNN Model for Freeway Incident Detection, *Journal of Transportation Engineering* **126**(6) (2000) 464–471.
3. H. Adeli and C. Yeh, Perceptron Learning in Engineering Design, *Microcomputers in Civil Engineering* **4**(4) (1989) 512–518.
4. M. Ahmed, R. Riyaz and S. Afzal, A comparative study of various approaches for dialogue management, *International Journal of Advanced Computer Technology* **2**(4) (2013) 89–96.
5. A. Alexandridis, Evolving RBF neural networks for adaptive soft-sensor design, *International Journal of Neural Systems* **23**(06) (2013).
6. 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)** (2008) 1249–1259.
7. B. Baruque, E. Corchado and H. Yin, The S2-Ensemble fusion algorithm, *International Journal of Neural Systems* **21(06)** (2011) 505–525.
 8. T. Bickmore, K. Puskar, E. Schlenk, L. Pfeifer and S. Sereika, Maintaining reality: Relational agents for antipsychotic medication adherence, *Interacting with Computers* **22** (2010) 276–288.
 9. A. Black, S. Burger, B. Langner, G. Parent and M. Eskenazi, Spoken dialog challenge 2010, *Proc. of Spoken Language Technology Workshop (SLT'10)*, (Berkeley, USA, 2010), pp. 448–453.
 10. D. Bohus and A. Rudnicky, The RavenClaw dialog management framework: Architecture and systems, *Computer Speech and Language* **23(3)** (2009) 332–361.
 11. I. Boulkabeit, L. Mthembu, F. D. L. Neto and T. Marwala, Finite Element Model Updating Using Fish School Search and Volitive Particle Swarm Optimization, *Integrated Computer-Aided Engineering* **22(4)** (2014) 361–376.
 12. J. Bradley, D. Benyon, O. Mival and N. Webb, Wizard of Oz experiments and companion dialogues, *Proc. of 24th BCS Interaction Specialist Group Conference (BCS'10)*, (Dundee, UK, 2010), pp. 117–123.
 13. J. Carvalho, F. Batista and L. Coheur, A Critical Survey on the use of Fuzzy Sets in Speech and Natural Language Processing, *Proc. of IEEE World Congress on Computational Intelligence (WCCI'12)*, (Brisbane, Australia, 2012), pp. 270–277.
 14. E. Castillo, D. Peteiro-Barral, B. G. Berdinas and O. Fontenla-Romero, Distributed One-class Support Vector Machine, *International Journal of Neural Systems* **25(7)** (2015).
 15. C. Chen, N. Liu, K. Chang and G. Su, Dynamic Gesture Recognition Based on Fuzzy Neural Network Classifier, *Proc. of 6th Conference on Advances in Computer-Human Interactions (ACHI'13)*, (Nice, France, 2012), pp. 57–61.
 16. L. Coletta, E. Hruschka, A. Acharya and J. Ghosh, Using metaheuristics to optimize the combination of classifier and cluster ensembles, *Integrated Computer-Aided Engineering* **22(3)** (2015) 229–242.
 17. P. Crook, S. Keizer, Z. Wang, W. Tang and O. Lemon, Real user evaluation of a POMDP spoken dialogue system using automatic belief compression, *Computer Speech and Language* **28** (2014) 873–887.
 18. H. Cuayáhuitl, S. Renals, O. Lemon and H. Shimodaira, Human-Computer Dialogue Simulation Using Hidden Markov Models, *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'05)*, (San Juan, Puerto Rico, 2005), pp. 290–295.
 19. N. Dethlefs and H. Cuayáhuitl, Hierarchical reinforcement learning for situated natural language generation, *Natural Language Engineering* **21(3)** (2015) 391–435.
 20. R. Elakkiya, K. Selvamani, S. Kanimozhi, R. Velumadhava and J. Senthilkumar, An interactive system for sensory and gustatory impaired people based on hand gesture recognition, *Procedia Engineering* **38** (2012) 3166–3172.
 21. K. Engelbrecht, *Estimating Spoken Dialog System Quality with User Models* (Springer, 2012).
 22. S. Espana, F. Zamora, M. Castro and J. Gorbe, Efficient bp algorithms for general feedforward neural networks, *Lecture Notes in Computer Science* **4527** (2007) 327–336.
 23. G. D. Fabbriozio, G. Tur, D. Hakkani-Tür, M. Gilbert, B. Renger, D. Gibbon, Z. Liu and B. Shahraray, Bootstrapping spoken dialogue systems by exploiting reusable libraries, *Natural Language Engineering* **14(3)** (2008) 313–335.
 24. E. Ferreira and F. Lefevre, Reinforcement-learning based dialogue system for human-robot interactions with socially-inspired rewards, *Computer Speech and Language* **34** (2015) 256–274.
 25. J. Florido, H. Pomares and I. Rojas, Generating balanced learning and test sets for function approximation problems, *International Journal of Neural Systems* **21(03)** (2011) 247–263.
 26. K. Georgila, M. Wolters and J. Moore, Simulating the Behaviour of Older versus Younger Users when Interacting with Spoken Dialogue Systems, *Proc. of 46th Meeting of the Association for Computational Linguistics (ACL'08)*, (Columbus, Ohio, USA, 2013), pp. 499–504.

27. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal Component Analysis-Enhanced Cosine Radial Basis Function Neural Network for Robust Epilepsy and Seizure Detection, *IEEE Transactions on Biomedical Engineering* **55**(2) (2008) 512–518.
28. D. Giacobello, J. Wung, R. Pichevar and J. Atkins, Tuning methodology for speech enhancement algorithms using a simulated conversational database and perceptual objective measures, *Proc. of 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA'14)*, (Villers-les-Nancy, France, 2014), pp. 62–66.
29. D. Griol, L. Hurtado, E. Segarra and E. Sanchis, A Statistical Approach to Spoken Dialog Systems Design and Evaluation, *Speech Communication* **50**(8-9) (2008) 666–682.
30. D. Griol, J. Iglesias, A. Ledezma and A. Sanchis, A practical application of evolving fuzzy-rule-based classifiers for the development of spoken dialog systems, *Proc. of 10th International Conference on Artificial Intelligence Applications and Innovations (AIAI'14)*, (Rodhes, Greece, 2014), pp. 307–316.
31. K. Han, D. Yu and I. Tashev, Speech Emotion Recognition Using Deep Neural Network and Extreme Learning Machine, *Proc. of 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, (Singapore, 2014), pp. 223–227.
32. H. Hastie, N. Merigaud, X. Liu and O. Lemon, “Let’s Go, DUDE!” Using the Spoken Dialogue Challenge to teach Spoken Dialogue development, *Proc. of IEEE Spoken Language Technology Workshop (SLT'10)*, (Berkeley, USA, 2010), pp. 466–471.
33. P. Heeman, Combining reinforcement learning with information-state update rules, *Proc. of 8th Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'07)*, (Rochester, New York, USA, 2007), pp. 268–275.
34. T. Heinroth and W. Minker, *Introducing Spoken Dialogue Systems into Intelligent Environments* (Kluwer Academic Publishers, 2012).
35. M. Henderson, B. Thomson and S. Young, Deep Neural Network Approach for the Dialog State Tracking Challenge, *Proc. of 14th SIGdial Meeting on Discourse and Dialogue (SIGdial'13)*, (Metz, France, 2013), pp. 467–471.
36. R. Higashinaka, K. Sudoh and M. Nakano, Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems, *Speech Communication* **48** (2006) 417–436.
37. G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath and B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition, *Signal Processing Magazine* **4527** (2013) 327–336.
38. H. Hofmann, A. Silberstein, U. Ehrlich, A. Berton, C. Muller and A. Mahr, *Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice* (Springer, 2014), ch. Development of Speech-Based In-Car HMI Concepts for Information Exchange Internet Apps, pp. 15–28.
39. O. Horchak, J.-C. Giger, M. Cabral and G. Pochwatko, From demonstration to theory in embodied language comprehension: A review, *Cognitive Systems Research* **29-30** (2014) 66–85.
40. S. Hung and H. Adeli, Parallel Backpropagation Learning Algorithms on Cray Y-MP8/864 Supercomputer, *Neurocomputing* **5**(6) (1993) 287–302.
41. S. Hung and H. Adeli, A Parallel Genetic/Neural Network Learning Algorithm for MIMD Shared Memory Machines, *IEEE Transactions on Neural Networks* **5**(6) (1994) 900–909.
42. S. Hung and H. Adeli, Object-Oriented Back Propagation and Its Application to Structural Design, *Neurocomputing* **6**(1) (1994) 45–55.
43. J. Iglesias, P. Angelov, A. Ledezma and A. Sanchis, Evolving Classification of Agents’ Behaviours: A General Approach, *Evolving Systems* **1**(3) (2010) 161–171.
44. J. Iglesias, P. Angelov, A. Ledezma and A. Sanchis, Creating Evolving User Behavior Profiles Automatically, *IEEE Transactions on Knowledge Data Engineering* **24**(5) (2012) 854–867.
45. K. Jackowski, B. Krawczyk and M. Wozniak, Improved Adaptive Splitting and Selection: the Hybrid Training Method of a Classifier Based

- on a Feature Space Partitioning, *International Journal of Neural Systems* **24**(3) (2014).
46. S. Jung, C. Lee, K. Kim, D. Lee and G. Lee, Hybrid user intention modeling to diversify dialog simulations, *Computer Speech and Language* **25**(2) (2011) 307–326.
 47. F. Jurcicek, B. Thomson and S. Young, Reinforcement learning for parameter estimation in statistical spoken dialogue systems, *Computer Speech and Language* **26** (2012) 168–192.
 48. A. Karim and H. Adeli, Comparison of the Fuzzy - Wavelet RBFNN Freeway Incident Detection Model with the California Algorithm, *Journal of Transportation Engineering* **128**(1) (2002) 21–30.
 49. A. Karim and H. Adeli, Radial Basis Function Neural Network for Work Zone Capacity and Queue Estimation, *Journal of Transportation Engineering* **129**(5) (2003) 494–503.
 50. S. Kartakis, A Design-and-Play Approach to Accessible User Interface Development in Ambient Intelligence Environments, *Journal Computers in Industry* **61**(4) (2010) 318–328.
 51. D. Kim, S. Rho and E. Hwang, Classification and Indexing Scheme of Large-Scale Image Repository for Spatio-temporal Landmark Recognition, *Integrated Computer-Aided Engineering* **22**(2) (2015) 201–213.
 52. Y. Kimura, K. Araki, Y. Momouchi and K. Tochinal, Spoken dialogue processing method using inductive learning with genetic algorithm, *Systems and Computers in Japan* **35**(12) (2004) 67–82.
 53. V. Kodogiannis, M. Amina and I. Petrounias, A clustering-based fuzzy wavelet neural network model for short-term load forecasting, *International Journal of Neural Systems* **23**(05) (2013).
 54. K. Kopp, M. Britt, K. Millis and A. Graesser, Improving the efficiency of dialogue in tutoring, *Learning and Instruction* **22**(5) (2012) 320–330.
 55. M. Kotti, E. Benetos, C. Kotropoulos and I. Pitas, A neural network approach to audio-assisted movie dialogue detection, *Neurocomputing* **71** (2007) 157–166.
 56. I. Lane, S. Ueno and T. Kawahara, Cooperative dialogue planning with user and situation models via example-based training, *Proc. of Workshop on Man-Machine Symbiotic Systems*, (Kyoto, Japan, 2004), pp. 2837–2840.
 57. R. Laroche, G. Putois, P. Bretier, S. Young and O. Lemon, Requirements Analysis and Theory for Statistical Learning Approaches in Automaton-Based Dialogue Management, tech. report, School of Informatics, Edinburgh University (2008).
 58. C. Lee, S. Jung, K. Kim, D. Lee and G. Lee, Recent Approaches to Dialog Management for Spoken Dialog Systems, *Journal of Computing Science and Engineering* **4**(1) (2010) 1–22.
 59. G. Lee, H. K. Kim, M. Jeong and J. Kim, *Natural Language Dialog Systems and Intelligent Assistants* (Springer, 2015).
 60. O. Lemon and O. Pietquin, *Data-Driven Methods for Adaptive Spoken Dialogue Systems. Computational Learning for Conversational Interfaces* (Springer, 2012).
 61. D. Li and D. Yu, Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing* **7**(3-4) (2014) 197–387.
 62. L. Li, J. Williams and S. Balakrishnan, Reinforcement learning for dialog management using least-squares Policy iteration and fast feature selection, *Proc. of International Conference on Spoken Language Processing (Interspeech'09)*, (Brighton, UK, 2009), pp. 2475–2478.
 63. P. Lison, Model-based bayesian reinforcement learning for dialogue management, *Proc. of International Conference on Spoken Language Processing (Interspeech'2013)*, (Lyon, France, 2013), pp. 457–461.
 64. Y. Liu and E. Shriberg, Does active learning help automatic dialog act tagging in meeting data, *Proc. of International Conference on Spoken Language Processing (Interspeech'2005)*, (Lisbon, Portugal, 2005), pp. 2777–2780.
 65. J. Lopes, M. Eskenazi and I. Trancoso, From rule-based to data-driven lexical entrainment models in spoken dialog systems, *Computer Speech and Language* **31** (2015) 87–112.
 66. R. López-Cózar, Z. Callejas and M. McTear, Testing the performance of spoken dialogue systems by means of an artificially simulated user, *Artificial Intelligence Review* **26** (2006) 291–323.
 67. J. Mariani and S. Rosset, *Natural Interaction with Robots, Knowbots and Smartphones:*

- Putting Spoken Dialog Systems into Practice* (Springer, 2012).
68. M. Martínez-Rojas, N. Marín and M. Vila, An Approach for the Automatic Classification of Work Descriptions in Construction Projects, *Computer-Aided Civil and Infrastructure Engineering* **30**(12) (2015).
 69. M. McTear and Z. Callejas, *Voice Application Development for Android* (Packt Publishing, 2013).
 70. R. Meena, G. Skantze and J. Gustafson, Data-driven models for timing feedback responses in a Map Task dialogue system, *Computer Speech and Language* **28** (2014) 903–922.
 71. F. Metzke, X. Anguera, E. Barnard, M. Davel and G. Gravier, Language independent search in MediaEval’s Spoken Web Search task, *Computer, Speech and Language* **28**(5) (2014) 1066–1082.
 72. W. Minker, T. Heinroth, P. Strauss and D. Zaykovskiy, *Human-Centric Interfaces for Ambient Intelligence* (Elsevier, 2010), ch. Spoken Dialogue Systems for Intelligent Environments, pp. 453–478.
 73. T. Misu, A. Raux, R. Gupta and I. Lane, Situated language understanding for a spoken dialog system within vehicles, *Computer Speech and Language* **34** (2015) 186–200.
 74. I. Montalvo, J. Izquierdo, M. Herrera and R. Pérez-García, Water Distribution System Computer-aided Design by Agent Swarm Optimization, *Computer-Aided Civil and Infrastructure Engineering* **29**(6) (2014) 433–448.
 75. C. Montero and K. Araki, Unsupervised Language Independent Genetic Algorithm Approach to Trivial Dialogue Phrase Generation and Evaluation, *Natural Language Processing and Information System. Lecture Notes in Computer Science* **4592** (2007) 388–394.
 76. E. Muller, S. Gunnemann, I. Assent and T. Seidl, Evaluating clustering in subspace projections of high dimensional data, *Proc. VLDB*, (Lyon, France, 2009), pp. 1270–1281.
 77. J. O’Shea, Z. Bandar and K. Crockett, A Multi-classifier Approach to Dialogue Act Classification Using Function Words, *LNCS* **7270** (2012) 119–143.
 78. R. Ota and M. Kimura, Proposal of open-ended dialog system based on topic maps, *Procedia Technology* **17** (2014) 122–129.
 79. T. Paek and E. Horvitz, Conversation as action under uncertainty, *Proc. of 16th Conference on Uncertainty in Artificial Intelligence*, (San Francisco, USA, 2000), pp. 455–464.
 80. T. Paek and R. Pieraccini, Automating spoken dialogue management design using machine learning: An industry perspective, *Speech Communication* **50**(8-9) (2008) 716–729.
 81. R. Pieraccini, *The Voice in the Machine: Building computers that understand speech* (MIT Press, 2012).
 82. J. Planells, L. Hurtado, E. Sanchis and E. Segarra, An Online Generated Transducer to Increase Dialog Manager Coverage, *Proc. of International Conference on Spoken Language Processing (Interspeech’12)*, (Portland, USA, 2012).
 83. A. Raux, B. Langner, A. Black and M. Eskenazi, Let’s go public! taking a spoken dialog system to the real world, *Proc. of International Conference on Spoken Language Processing (Interspeech’05)*, (Lisbon, Portugal, 2005), pp. 885–888.
 84. A. Raux, B. Langner, A. Black and M. Eskenazi, Doing research on a deployed spoken dialog system: One year of Lets Go! experience, *Proc. of International Conference on Spoken Language Processing (Interspeech’06)*, (Pittsburgh, Pennsylvania, USA, 2006), pp. 65–68.
 85. K. Reschke, A. Vogel and D. Jurafsky, Generating Recommendation Dialogs by Extracting Information from User Reviews, *Proc. of 51st Annual Meeting of the Association for Computational Linguistics (ACL’13)*, (Sofia, Bulgaria, 2013), pp. 499–504.
 86. S. Rodríguez, Y. de Paz, J. Bajo and J. M. Corchado, Social-based Planning Model for Multi-agent Systems, *Expert Systems with Applications* **38**(10) (2011) 13005–13023.
 87. D. E. Rumelhart, G. E. Hinton and R. J. Williams, *PDP: Computational models of cognition and perception, I* (MIT Press, 1986), ch. Learning internal representations by error propagation, pp. 319–362.
 88. Y. Said, *Handbook of Statistics, Vol. 24* (Elsevier, 2005), ch. On Genetic Algorithms and their Applications, pp. 359–390.
 89. J. Schatzmann, B. Thomson, K. Weilhammer,

- H. Ye and S. Young, Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System, *Proc. HLT/NAACL*, (Rochester, USA, 2007), pp. 149–152.
90. 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)** (2006) 97–126.
91. A. Schmitt, S. Ultes and W. Minker, A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let’s Go Bus Information System, *Proc. of 8th International Conference on Language Resources and Evaluation (LREC’12)*, (Istanbul, Turkey, 2012), pp. 3369–3375.
92. A. Schmitt, *Towards Adaptive Spoken Dialog Systems* (Springer, 2012).
93. D. Schnelle-Walka, S. Radomski, S. Radeck-Arneth and M. Muhlhauser, Towards an Information State Update Model Approach for Non-verbal Communication, *Lecture Notes in Computer Science* **8547** (2014) 226–230.
94. E. Selfridge, I. Arazmendi, P. Heeman and J. Williams, Integrating incremental speech recognition and POMDP-based dialog systems, *Proc. of 12th SIGdial Meeting on Discourse and Dialogue (SIGdial’12)*, (Seoul, South Korea, 2012), pp. 467–471.
95. S. Seneff, M. Adler, J. Glass, B. Sherry, T. Hazen, C. Wang and T. Wu, Exploiting Context Information in Spoken Dialogue Interaction with Mobile Devices, *Proc. of Int. Workshop on Improved Mobile User Experience (IMUX’07)*, (Toronto, Canada, 2007), pp. 1–11.
96. C. Seon, H. Kim and J. Seo, A statistical prediction model of speakers intentions using multi-level features in a goal-oriented dialog system, *Pattern Recognition Letters* **33** (2012) 1397–1404.
97. F. Shabbir and P. Omenzetter, Particle swarm optimization with sequential niche technique for dynamic finite element model updating, *Computer-Aided Civil and Infrastructure Engineering* **30(5)** (2015) 359–375.
98. N. Siddique and H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing* (Wiley, 2013).
99. M. Sidorov, C. Brester, W. Minker and E. Semenkin, Speech-Based Emotion Recognition: Feature Selection by Self-Adaptive Multi-Criteria Genetic Algorithm, *Proc. of 9th International Conference on Language Resources and Evaluation (LREC’14)*, (Reykjavik, Iceland, 2014), pp. 3481–3485.
100. A. Stent, S. Stenchikova and M. Marge, Reinforcement learning of dialogue strategies with hierarchical abstract machines, *Proc. of Spoken Language Technology Workshop (SLT’06)*, (Palm Beach, Aruba, 2006), pp. 210–213.
101. D. Suendermann and R. Pieraccini, One year of contender: what have we learned about assessing and tuning industrial spoken dialog systems?, *Proc. of NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data (SDCTD’12)*, (Montreal, Canada, 2012), pp. 45–48.
102. B. Thomson, K. Yu, S. Keizer, M. Gasic, F. Jurcicek, F. Mairesse and S. Young, Bayesian dialogue system for the Let’s Go Spoken Dialogue Challenge, *Proc. of IEEE Spoken Language Technology Workshop (SLT’10)*, (Berkeley, USA, 2010), pp. 460–465.
103. F. Torres, L. Hurtado, F. García, E. Sanchis and E. Segarra, Error handling in a stochastic dialog system through confidence measures, *Speech Communication* **45(3)** (2005) 211–229.
104. F. Torres, E. Sanchis and E. Segarra, User simulation in a stochastic dialog system, *Computer Speech and Language* **22(3)** (2008) 230–255.
105. M. Tsai, The VoiceXML dialog system for the e-commerce ordering service, *Proc. of 9th International Conference on Computer Supported Cooperative Work in Design (CSCWD’05)*, (Coventry, UK, 2005), pp. 95–100.
106. R. Vidal, Subspace clustering, *IEEE Signal Processing Magazine* **28(2)** (2011) 52–68.
107. R. Vipperla, M. Wolters and S. Renals, *Spoken dialogue interfaces for older people*, *Advances in Home Care Technologies* (IOS Press, 2012), pp. 118–137.
108. F. Wang and K. Swegles, Modeling user behavior online for disambiguating user input in a spoken dialogue system, *Speech Communication* **55** (2013) 84–98.
109. Y. Wilks, R. Catizone, S. Worgan and M. Turunen, Some background on dialogue management and conversational speech for dialogue

- systems, *Computer Speech and Language* **25** (2011) 128–139.
110. J. Williams, I. Arizmendi and A. Conkie, Demonstration of AT&T Let's Go: A production-grade statistical spoken dialog system, *Proc. of Spoken Language Technology Workshop (SLT'10)*, (Berkeley, California, USA, 2010), pp. 157–158.
 111. J. Williams, P. Poupart and S. Young, *Recent Trends in Discourse and Dialogue* (Springer, 2006), ch. Partially Observable Markov Decision Processes with Continuous Observations for Dialogue Management, pp. 191–217.
 112. J. Williams, The best of both worlds: Unifying conventional dialog systems and POMDPs, *Proc. of International Conference on Spoken Language Processing (InterSpeech'08)*, (Brisbane, Australia, 2008), pp. 1173–1176.
 113. J. Williams, Challenges and opportunities for state tracking in statistical spoken dialog systems: Results from two public deployments, *Journal of Selected Topics Signal Processing* **6**(8) (2012) 959–970.
 114. J. Williams, N. Niraula, P. Dasigi, A. Lakshmiratan, C. Garcia-Jurado, M. Reddy and G. Zweig, Rapidly scaling dialog systems with interactive systems, *Proc. of 6th International Workshop on Spoken Dialog Systems (IWSDS)*, (Busan, South Korea, 2015), pp. 227–231.
 115. A. Winterboer, M. Tietze, M. Wolters and J. Moore, The user model-based summarize and refine approach improves information presentation in spoken dialog systems, *Computer Speech and Language* **25** (2011) 175–191.
 116. J. Wu, J. Tseng and W. Tsai, A hybrid linear text segmentation algorithm using hierarchical agglomerative clustering and discrete particle swarm optimization, *Integrated Computer-Aided Engineering* **21**(1) (2014) 35–46.
 117. A. Yahya, R. Mahmod and A. Ramli, Dynamic Bayesian networks and variable length genetic algorithm for designing cue-based model for dialogue act recognition, *Computer Speech and Language* **24** (2010) 190–218.
 118. Y. Yang, Y. Li, Y. Gao, H. Yin and Y. Tang, Structurally Enhanced Incremental Neural Learning for Image Classification with Subgraph Extraction, *International Journal of Neural Systems* **24**(7) (2014).
 119. S. Young, The Statistical Approach to the Design of Spoken Dialogue Systems, tech. report, Cambridge University Engineering Department (UK) (2002).
 120. S. Young, M. Gasic, B. Thomson and J. Williams, POMDP-based statistical spoken dialogue systems: a review, *Proceedings of the IEEE* **101** (2013) 1160–1179.
 121. S. Young, J. Schatzmann, K. Weilhammer and H. Ye, The Hidden Information State Approach to Dialogue Management, *Proc. of 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Honolulu, Hawaii, USA, 2007), pp. 149–152.
 122. Y. Yu and T. McKelvey, A Robust Subspace Classification Scheme Based on Empirical Intersection Removal and Sparse Approximation, *Integrated Computer-Aided Engineering* **22**(1) (2015) 59–69.
 123. Z. Zeng, J. Xu, S. Wu and M. Shen, Antithetic Method-based Particle Swarm Optimization for a Queuing Network Problem with Fuzzy Data in Concrete Transportation Systems, *Computer-Aided Civil and Infrastructure Engineering* **29**(10) (2014) 771–800.
 124. Y. Zhang and W. Zhou, Multifractal Analysis and Relevance Vector Machine-based Automatic Seizure Detection in Intracranial, *International Journal of Neural Systems* **25**(6) (2015).
 125. I. Zukerman, S. Kim, T. Kleinbauer and M. Moshtaghi, Employing distance-based semantics to interpret spoken referring expressions, *Computer Speech and Language* **34** (2015) 154–185.