# A proposal for the development of adaptive spoken interfaces to access the Web

David Griol [a,*], José Manuel Molina [a], Zoraida Callejas [b]

[a] *Applied Artificial Intelligence Group, Department of Computer Science, Carlos III University of Madrid, Spain*
[b] *Spoken and Multimodal Dialogue Systems Group, Department of Languages and Computer Systems, University of Granada, Spain*

Spoken dialog systems have been proposed as a solution to facilitate a more natural human machine interaction. In this paper, we propose a framework to model the user's intention during the dialog and adapt the dialog model dynamically to the user needs and preferences, thus developing more efficient, adapted, and usable spoken dialog systems. Our framework employs statistical models based on neural networks that take into account the history of the dialog up to the current dialog state in order to predict the user's intention and the next system response. We describe our proposal and detail its application in the Let's Go spoken dialog system.

## 1. Introduction

Continuous advances in the development of information tech nologies and the miniaturization of devices have made it possible to access information, web services, and artificial intelligence systems from anywhere, at anytime and almost instantaneously through wireless connections [1]. Although devices such as smart phones and tablets are widely used today to access the web, the reduced size of the screen and keyboards makes the use of traditional graphical user interfaces (GUIs) difficult, especially for motor handicapped and visually impaired users. This way, alth ough mobile phones are designed to provide ubiquitous access to Internet, the present challenge is to make the enormous web content accessible to all mobile phone users by means of more natural communication metaphors.

Dialog systems go a step beyond GUIs by adding the possibility to communicate with these devices through other interaction modes such as speech [2 5]. These systems can be defined as computer programs designed to emulate human communication capabilities including several communication modalities. 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 speech recognition is to obtain the sequence of words uttered by a speaker [6]. Once the speech recognizer has provided an output, the system must understand what the user said. The goal of spoken language understanding is to obtain the semantics from the recognized sentence [7]. This process generally requires morphological, lexical, syntactical, semantic, discourse and pragmatical knowledge.

The dialog manager decides the next action of the system, interpreting the incoming semantic representation of the user input in the context of the dialog [8]. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and under standing errors, retrieves information from data repositories, and decides about the next system's response. In order to complete these tasks and decide “what to say”, the dialog manager needs to track the dialog history and update some representation of the current state of the dialog. In addition, the DM needs a dialog model that defines the conversational behavior of the system, for example when to take the initiative in a dialog, when to confirm a piece of information, how to identify and recover from recognition and understanding errors, and so forth.

Natural language generation is the process of obtaining sen tences in natural language from the non linguistic, internal repre sentation of information handled by the dialog system [9]. Finally, the TTS module transforms the generated sentences into synthe sized speech [10].

During the last years, Internet is playing an increasingly important role for making speech technology available anywhere.

By giving the user the chance to interact with the web via natural language, users are provided with the possibility to come up with less restricted input. Initial application domains included simple solutions to provide a vocal interface to an existing web browser [11] or to access information in limited on line domains [12].

The performance of spoken language dialog systems to access web contents has improved over time, extending these initial application domains to more complex information retrieval and question answering applications [13,14], e commerce systems [15], surveys applications [16], recommendations systems [17], e learning and tutoring systems [18,19], in car systems [20,21], remote control of devices and robots in smart environments [22,23], healthcare and Ambient Assisted Living systems [24,25], or embodied dialog systems and companions [26,27].

Many companies and public institutions have also taken advan tage of using Voice Portals as a cheap and effective way of supporting customers [28]. Besides spoken and written language have become popular with the incorporation of chatbots for web based customer support [29]. The spread of information through social web media has also made possible to generate models for conversation that take profit of the data's vast size and conversational nature of web applications like Twitter or Wikipedia [30,31], and also allow spoken interaction in 3 D immersive virtual environments like Second Life or Open Simulator [32,33]. Mobile devices have also extended the possibilities of integrating speech interaction to develop advanced apps that access the web [2].

The described systems are usually designed ad hoc for their specific domain using rule based 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 user requirements or deal with new tasks is a time consuming process that implies a considerable effort, with the ever increasing problem of dialog complexity [34,35].

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 [36,37]. 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 com munication 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 appli cation of statistical approaches to user modeling and dialog man agement makes it possible to consider a wider space of dialog strategies in comparison to engineered rules [38,8].

The main reason is that statistical approaches for dialog man agement can be trained from real dialogs, modeling the variability in user behaviors. Although the parameterization of the model depends on expert knowledge of the task, the final objective is to develop dialog systems that have a more robust behavior, better portability, and are easier to adapt to different user profiles or tasks [39]. This would help to create user adapted speech enabled interfaces for the wide range of web based applications previously described, reducing the effort and time required by hand crafted designed systems to consider specific users requirements or deal with new tasks with the ever increasing problem of dialog com plexity [34,35].

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 make the estimation of the dialog model from the training data and practical domains manageable.

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 manage ment. 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. In this section we also discuss the evaluation results obtained in this application. Finally, in Section 5 we present the conclusions and outline guidelines for future work.

## 2. Related work

The design and development of a comprehensive adaptive spoken dialog system can be conceptually composed of two inter connected 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 [39 41]. 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 [39].

Intentions cannot be observed, but they can be described using the speech act and dialog act theories [42,43]. The notion of a dialog act plays a key role in studies of dialog, in particular in the interpretation of the communicative behavior of the participants; in building annotated dialog corpora; and in the design of dialog management systems for spoken human computer dialog. A dialog act has two main components: a communicative function and a semantic content. A standard representation for dialog act annotation is proposed in [44], which uniformizes the semantic annotation of dialog corpora. Thus, it provides a standard repre sentation for the output provided by the SLU module in dialog systems and its communication with the dialog manager (e.g., *Yes No Question*, *Reject*, *Conventional Closing*, or *Thanks*).

In recent years, simulation on the intention level has been most popular [39]. This approach was first used by [45] and has been adopted for user simulation by most research groups [46 49]. Modeling interaction on the intention level avoids the need to reproduce the enormous variety of human languages on the level of speech signals [50,51] or word sequences [52,53].

In [54], Eckert, Levin and Pieraccini introduce the use of statistical models to predict the user's intention by means of a n gram model, predicting the user action that is most probable given the dialog history of system and user actions. In practice, data sparsity makes long dialog histories intractable in practice. Eckert et al. approximate the full history with a bigram model. Bigram and in general n grams models consider that the 2 or N previous user responses are informative enough to predict the next one, an assumption which is usually valid but sometimes leads to lose information which is necessary for a correct compu tation of the user model. The main weakness is that the responses generated may correspond well to the previous system action, but do not make sense in the larger context of the dialog.

In [46], Scheffler and Young propose a graph based model. In depth knowledge of the task and great manual effort are necessary for the specification of all possible dialog paths (i.e., different conversations defined as sequences of user and system dialog turns). The arcs of the graph represent actions and the nodes represent “choice points”. Some of these points are identified as probabilistic choice points (i.e., random decision by a simulated user estimated from training data). The remaining nodes are deterministic choice points. Pietquin, Beaufort and Dutoit combine characteristics of Scheffler and Young's model and Levin's model to reduce the manual effort necessary for the construction of such graphs [55].

Georgila, Henderson and Lemon 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 [48]. Dialog is viewed as a sequence of Information States [56], 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. Cuayáhuitl et al. define a HMMs based dialog simulation technique in which both the user and system behaviors are simulated [49]. Instead of training only a generic HMM model to simulate any type of dialog, a submodel is trained for each one of the objectives.

A data driven user intention simulation method that integrates diverse user discourse knowledge (cooperative, corrective, and self directing) is presented in [57]. 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 [58]. 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 fea tures, and domain level features as inputs to a statistical model based on the Maximum Entropy Model (MEM) [59]. 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.

Very recently, Wang and Swegles propose a technique that employs knowledge about the user's activity to disambiguate their spoken inputs [60]. A Reinforcement Learning algorithm is pro posed 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.

In [61], the authors present a technique for user simulation based on explicit representations of the user goal and the user agenda. This model formalizes human machine dialogs at a seman tic level as a sequence of states and dialog acts for which the user has a predefined plan (agenda) that may vary during the conversa tion. 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, 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. Although dialog manage ment is only a part of the development cycle of spoken dialog systems, it can be considered one of the most demanding tasks given that this module encapsulates the logic of the speech application [62]. This way, the design of an appropriate dialog management strategy is at the core of dialog system engineering. A comprehensive study of dialog management methodologies and architectures is presented in [63 65].

Automating dialog management by means of statistical meth odologies is useful for developing, deploying and re deploying applications and also reducing the time consuming process of hand crafted design. Statistical models can be trained with cor pora 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 [39]. The goal is to build systems that exhibit more robust performance, improved portability, better scalability and easier adaptation to other tasks.

The most widespread methodology for machine learning of dialog strategies consists of modeling human computer interac tion as an optimization problem using Markov Decision Processes (MDP) and reinforcement methods [45,66]. The main drawback of this approach is that the large state space of practical domains makes its direct representation intractable [67]. Partially Obser vable MDPs (POMDPs) outperform MDP based dialog strategies since they provide an explicit representation of uncertainty [68]. However, they are also limited to small scale problems, since the state space would be huge and exact POMDP optimization is again intractable [67].

Other authors have combined conventional dialog managers with a fully observable Markov decision process [69,70], or proposed the use of multiple POMDPs and selecting actions using hand crafted rules [71]. In [72], the authors combine the robustness of the POMDP with conventional approaches. The POMDP then chooses the best action from this limited set. Bayesian reinforcement learning frameworks for learning the POMDP parameters from data have been recently proposed in [73,74]. Other interesting approaches for statistical dialog management are based on modeling the system by means of Hidden Markov Models [49], stochastic Finite State Transducers [75 77], or Bayesian Networks [78,79].

Our methodology for dialog management (Section 3) 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 informa tion 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.

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 artifi cial neural networks [8,80,81]. The best results were obtained using a multilayer perceptron (MLP) [82]. Neural networks have also proven to be useful in for other tasks related to natural language processing [83,84], such as the estimation of text sim ilarity [85,86], handwritten text recognition [87], automatic lan guage recognition [88], out of vocabulary word detection [89], word sense disambiguation [90], associative memory models [91], language model estimation for speech recognition [92], spoken language understanding [93], or question answering [94].

## 3. Proposed framework to develop adaptive spoken dialog systems

Fig. 1 shows the architecture that integrates our proposed framework to generate adaptive spoken dialog systems. A User Modeling module considers the previous dialog interactions and specific features of the user (defined by means of user profiles) to predict the user intention, defined as the next user action, which we represent by one or more dialog acts as described in the previous section.

The Dialog Manager takes as an 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). The following subsections describe the statistical methodologies pro posed for the development of the two modules.

### 3.1. User modeling

Our proposed technique for user modeling simulates the user intention providing the next user dialog act in the same repre sentation 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 [8]. This way, each dialog is represented by

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

where $A_1$ is the greeting turn of the system, and $U_n$ is the last user turn. 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 cate gories. 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., organiza tion, person or location names), a time identifier (e.g., dates, time expressions or durations), or quantities and numerical expressions (e.g., monetary values, percentages or phone numbers).

Our model is based on the one proposed in [95]. 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 time table 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 label of the $i$th turn, and $ST_i^u$ denotes the subtask label to which the $i$th turn contributes. The dialog act is determined from the information about the turn and the previous dialog context (i.e., $k$ previous utterances) as shown in the following equation:

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

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

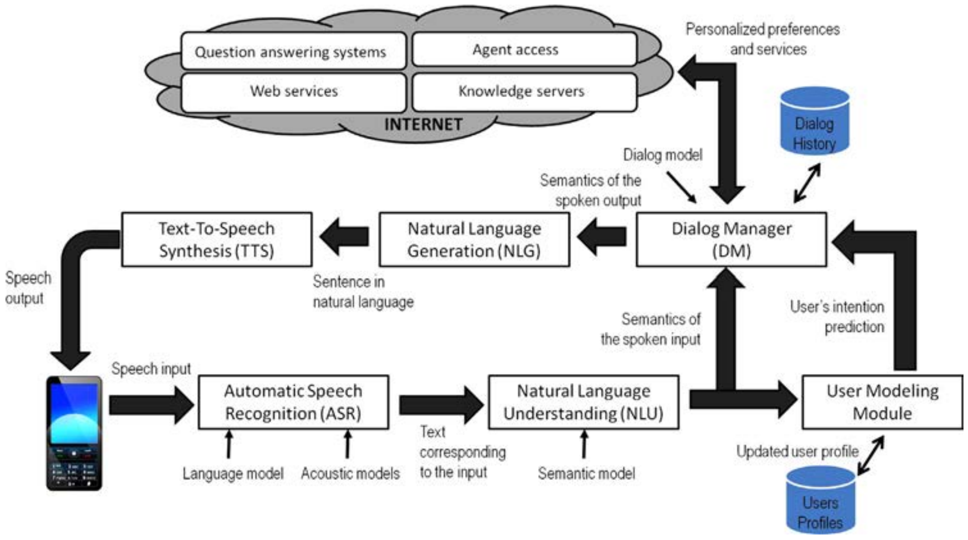
Question answering systems
Agent access
Web services
Knowledge servers
INTERNET
Personalized preferences and services
Dialog History
Dialog model
Semantics of the spoken output
Text-To-Speech Synthesis (TTS)
Natural Language Generation (NLG)
Dialog Manager (DM)
Speech output
Sentence in natural language
Semantics of the spoken input
User's intention prediction
Speech input
Automatic Speech Recognition (ASR)
Natural Language Understanding (NLU)
User Modeling Module
Text corresponding to the input
Language model
Acoustic models
Semantic model
Updated user profile
Users Profiles

**Fig. 1.** Architecture to develop adaptive spoken dialog systems.

dialog context, as shown in the following equation:

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

In our proposal, we consider static and dynamic features to estimate the conditional distributions shown in (Eqs. (1) and 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, predicate arguments in each utterance, and also a set of features included in a user profile. 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 user profile is composed of the following user features:

- Id, which they can use to log in to the system.
- Gender.
- Experience, which can be either 0 for novel users (first time the user calls the system) or the number of times the user has interacted with the system.
- Skill level, estimated taking into account the level of expertise, the duration of their previous dialogs and the time that was necessary to access a specific content and the date of the last interaction with the system. A low, medium, high or expert level is assigned using these measures. The experience and skill level features are subtly different. The former encodes the number of previous interactions, while the latter indicates the difficulty experienced by the user to succeed in such interactions.
- Most frequent objective of the user.
- Reference to the physical location of the user during previous interactions (when available), in which the previous set of objective and subjective parameters for the user has been estimated.

As described in [95], the conditional distributions shown in (Eqs. (1) and 2) can be estimated by means of the general technique of choosing the maximum entropy (MaxEnt) distribu tion that properly estimates the average of each feature in the training data [96]. This can be written as a Gibbs distribution parameterized with weights $\lambda$ 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 $\Phi$ denotes the vector of the described features used for user modeling

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

Such calculation outperforms other state of the art approaches [95,97,96], 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 follows:

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

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

## *3.2. Dialog management*

A conventional dialog manager maintains the dialog state encoded as a form or frame and uses two functions for selecting the next system response, that we will denote by $G$ and $F$. For a given dialog state $n$, $G(n)$ decides which system action ($a$) to output, and then after observation $o$ has been received, $F(n, o)$ decides how to update the dialog state $n$ to $n_0$ ($n$ after observing $o$). This process repeats until the dialog ends.

In a statistical approach, the conventional dialog manager is extended in three respects: firstly, its action selection function $G(n)$ is changed to output a set of one or more ($M$) valid actions given a dialog state $n$, $G(n) = \{a_1, a_2, \ldots, a_M\}$. Next, its transition function $F$ is extended to allow for different transitions depending on which of the actions was taken, $F(n, a, o)$. This way, rather than maintaining a single hypothesis for the dialog state, these func tions maintain a distribution over many hypotheses, which are ordered from the most to the least probable, for the correct dialog state. Then, the dialog manager may select to consider only the best or a set of the $N$ best options available.

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

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

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:

- Sequence of user dialog acts provided by the user throughout the previous history of the dialog (i.e., the output of the SLU module).
- Predicted user dialog act (generated by means of Eq. (1)).
- Predicted user subtask (generated by means of Eq. (2)).

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 the following equation:

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

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 $\text{IR}_{i \quad 1}$. Hence, for the selection of the best system response $A_i$, we take into account the *IR* from turn 1 to turn $i \quad 1$, 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 [98].
- (2) The concept or attribute has a confidence score that is lower than the given threshold.

We propose to solve (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. As stated in Section 2, the classification function can be defined in several ways. The best results were obtained using a multilayer perceptron (MLP) where the input layer receives the current state of the dialog, which is represented by the term ($\mathrm{IR}_{i-1}, A_i$). The values of the output layer can be viewed as the a posteriori probability of selecting the different system responses given the current situation of the dialog.

Fig. 2 summarizes the combination of the proposed user modeling and dialog management methodologies. As can be observed, the user modeling module provides a prediction of the next user dialog act and the current subtask of the dialog. The set of user dialog acts and predicted values for the current user's dialog act and subtask are used to update the interaction register. The dialog manager considers this register and the last system response for the selection of the next system action.

### 3.3. MLP classifier

In order to apply a MLP to select the system answer, as previously stated, the input layer 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 ($C$)

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

- Interaction register ($\mathrm{IR}_{i-1}$): 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 ($N$). The prediction of the current dialog subtask is modeled using a variable, which has as many bits as possible subtasks ($T$)

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3, \quad i = 2, \ldots, N+1$$

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

$$\vec{x}_{N+3} = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_T}) \in \{0,1\}^T$$

- 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

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0,1\}^3, \quad i = N+4, \ldots, N+6$$

For the process of classification, the number of output units of the MLP is defined as the number of classes, $|C|$, and the input layer must hold the input samples $(IR_{i-1}, S_{i-1})$. For uniclass samples, the activation level of an output unit in the MLP can be interpreted as an approximation of the a posteriori probability that the input sample belongs to the corresponding class [82,84]. Therefore, given an input sample $x$, the trained MLP computes $g_c(\mathbf{x}, \omega)$ (the $c$th output of the MLP with parameters $\omega$ given the input sample $\mathbf{x}$), which is an approximation of the a posteriori probability $P(c|\mathbf{x})$. Thus, for MLP classifiers we can use the uniclass classification rule as

$$\hat{c} = \underset{c \in \mathcal{C}}{\operatorname{argmax}}\, P(c|\mathbf{x}) \approx \underset{c \in \mathcal{C}}{\operatorname{argmax}}\, g_c(\mathbf{x}, \omega)$$

where the variable $x$, which holds for the pair $(IR_{i-1}, S_{i-1})$, can be represented using the vector of characteristics

$$\vec{x} = \left(\vec{x}_1, \vec{x}_2, \vec{x}_3, \ldots, \vec{x}_{N+6}\right)$$

## 4. Application to the Let's Go dialog system

Let's Go is a spoken dialog system developed by the Carnegie Mellon University to provide bus schedule information in Pitts burgh. The system has had many users since it was made available for the general public in 2005 20,000 calls collected just from March to December of 2005 [99], so there is a substantial dataset

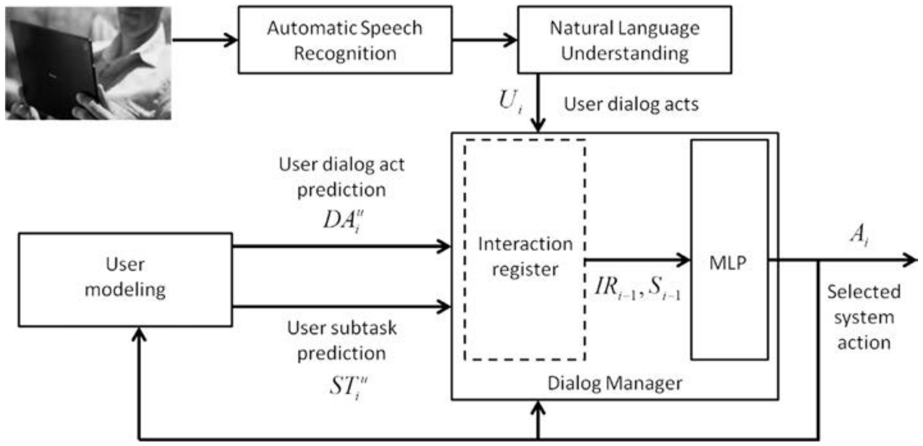

**Fig. 2.** Combination of the proposed user modeling and dialog management methodologies for the development of adaptive spoken dialog systems.

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 testers.

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 initiative [100]. The aim of the Challenge was to bring together multiple implementations of the same dialog task and deploy them in uncontrolled real user conditions and then make the results available for common evaluation techniques.

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 operative 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 experi mentation 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 [101,100,102,103].

Fig. 3 shows an example of a dialog extracted from the Let's Go corpus [103]. In each dialog, the user needs to provide a place of departure, a destination and a departure time. To model this, the system uses a set of user dialog acts that has been classified into 16 categories following the criteria described in [104]. Table 1 shows the defined categories of user dialog acts and their specific values. Four of the concepts are used to model where the user is leaving from (monument, pair of road names, neighborhood, or stop). The four concepts used for modeling the place of arrival are similar. Six concepts 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 concept *disc* models how the user issues "discourse" actions, which relate to only one tum in a dialog.

The 36 system 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), *statusre ports* (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).

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_re sults*, *provide_instructions*, *query_error*, and *goodbye*).

A total number of 22 features define the pair $(IR_{i-1}, S_{i-1})$ for the Let's Go task: the last system response $(A_{i-1})$, 18 features corresponding to the Interaction Register $IR_{i-1}$ (predicted current user dialog act, 16 task dependent user dialog acts, and predicted current dialog subtask), and 3 features corresponding to the task independent information (*Affirmation*, *Negation*, and *Not Understood* dialog acts). This information is the input to our MLP and the number of features is similar to other application domains in which our proposal has been previously applied [8].

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. As described in Table 1, 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.

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

**Table 1**
User dialog acts categories defined in the Let's Go system [104].

| Category | Size | Example values |
|---|---|---|
| *from* | 3 | ftstop, ftmonument, ftneigh |
| *fstop* | 328, 455 | "FORBES&MURRAY", "ANYWHERE&FORBES" |
| *fmon* | 52 | "AIRPORT", "CENTURY SQUARE" |
| *fneigh* | 220 | "DOWNTOWN", "SQUIRREL HILL" |
| *to* | 3 | ttstop, ttmonument, ttneigh |
| *tstop* | 328, 455 | "FORBES&MURRAY", "ANYWHERE&FORBES" |
| *tmon* | 52 | "AIRPORT", "CENTURY SQUARE" |
| *tneigh* | 220 | "DOWNTOWN", "SQUIRREL HILL" |
| *time* | 2 | "NEXT", time specific |
| *hour* | 12 | "ONE", "TWELVE" |
| *min* | 60 | "ZERO", "TEN", "THIRTY FIVE" |
| *pd* | 2 | "AM", "PM" |
| *day* | 2 | day 10, "TODAY", "WEDNESDAY", "TOMORROW" |
| *tref* | 4 | "ARRIVE BEFORE", "LEAVE AFTER" |
| *meth* | 4 | "RESTART", "FINISHED", constraints, |
| *disc* | 9 | "REPEAT", "FOLLOWING", "PREVIOUS", none |

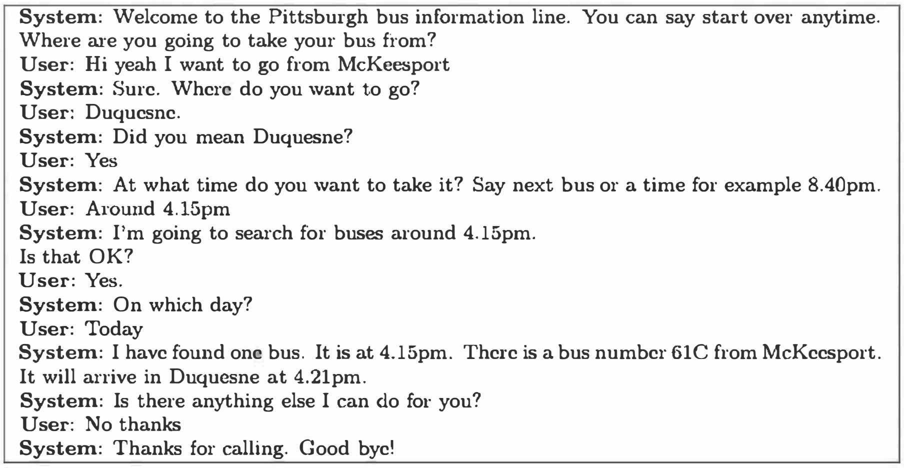
**System:** Welcome to the Pittsburgh bus information line. You can say start over anytime. Where are you going to take your bus from?
**User:** Hi yeah I want to go from McKeesport
**System:** Sure. Where do you want to go?
**User:** Duquesne.
**System:** Did you mean Duquesne?
**User:** Yes
**System:** 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!

Fig. 3. Example of dialog extracted from the Let's Go corpus.

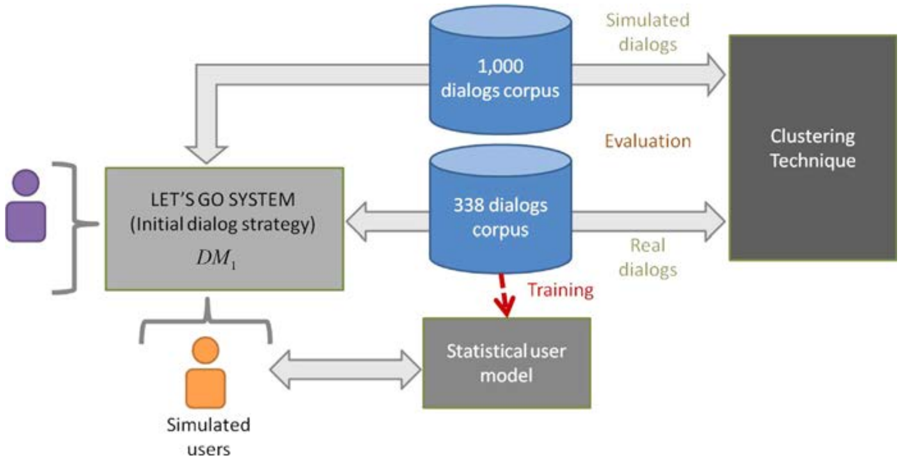

**Fig. 4.** Setup for the first experiment.

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, and the second experiment evaluates the appropriateness of the decision of the dialog manager computed considering the user model.

Thus, the main objective of the first experiment, described in Fig. 4, is to assess the proposed methodology for user modeling (Section 3.1). 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 1000 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. Hence, in the experiment we had to check whether the simulated dialogs generated with our user modeling technique were similar to those that would take place with real users.

Under this assumption, assessing the user model usually con sists in a subjective assessment of the realism of the simulated user (e.g. with human judges in a kind of Turing test) 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 (which are discussed in Section 4.1). Then, we employed a technique called subspace clustering [105] 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 dimensionality of the space beforehand unlike other state of the art approaches, which would lead to a loss of valuable information. The results of the first experiment are discussed in Section 4.1.

Fig. 5 shows the second experiment, in which we evaluated the operation of the complete framework, that is, with this experi ment 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 1817 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.

From our previous work on statistical dialog management [8], in this case 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 input 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 the following:

- *Matching*: the percentage of responses provided by $DM_2$ that are equal to the reference answer in the corresponding turn of the test corpus.
- *Coherence*: the percentage of answers provided by $DM_2$ that are coherent with the current state of the dialog although they are not necessarily the same that the reference answer.
- *Error*: the percentage of answers provided by $DM_2$ 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. The results of the second experiment are discussed in Section 4.2.

## *4.1. Evaluation of the user modeling module*

Table 2 shows the features computed for the 1338 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 [106], an implementation of the algorithm that can be integrated into the Weka machine learning tool.

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

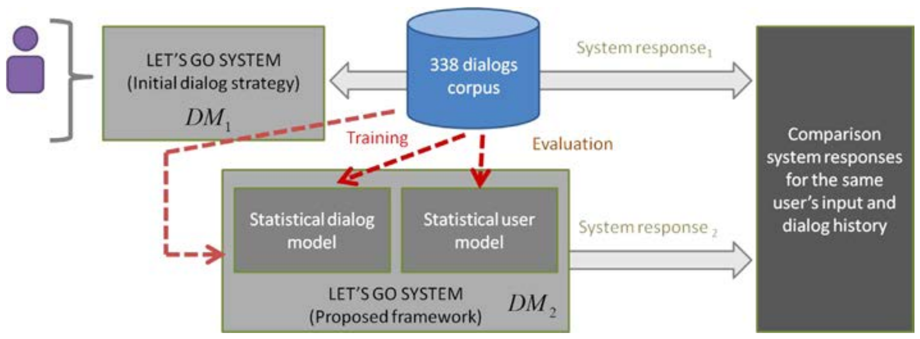

**Fig. 5.** Setup for the second experiment.

**Table 2**
Features computed for the simulated and real dialogs.

| | |
|---|---|
| *numExchanges* | Number of exchanges between user and system. An exchange is composed 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) |
| *percConfirm* | Percentage of confirmation dialog acts |
| *meanReasks* and *maxReasks* | mean and maximum re-asks |
| *numNoMatches*, *percNoMatches* | Number and percentage of no-matches |
| *numNoInputs*, *percNoInputs* | Number and percentage of no-inputs |
| *meanAVPs* | 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 *next bus*, 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 |
| *successDial* | Task success, determined automatically by checking if the objective of the dialog was reached |
| *SysActPerUserAct* | Number of system turns in a dialog divided by the total number of concepts provided by the user |

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 stan dard 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. Cluster 1 is composed 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 composed 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.

The users of the Let's Go system follow different objectives, e.g. some users simply look for a line connecting 2 stops, others need departure times, and others need information about complete connections. Completing these tasks with the system can involve navigation within the results, and in some cases also changing the query, e.g. if two stops are not connected by a line. Cluster 1 mainly comprises short dialogs, and cluster 2 contains searches for a connection between 2 stops. Dialogs with more than 2 stops are distributed across all clusters except cluster 1. Thus, clustering can be used to group dialogs with many different objectives to a few groups.

Fig. 6 shows the percentages of real and simulated dialogs in each cluster. It can be observed that both datasets are differently structured with regard to the distribution of clusters. Fig. 6 (down) shows that both are mainly in clusters 1 and 2. As those clusters contain more standard interactions, it seems that the user model was able to render such behaviors appropriately, which represent

**Table 3**
Results of the subspace clustering for the Let's Go task.

| Cluster (dimensions): *relevant features* | #Dialogs |
|---|---|
| 0 (4D): *numNoInputs*, *percNoInputs*, *successDial*, *percResults* | 59 |
| 1 (4D): *percNoInputs*, *meanAVP*, *numError*, *percError* | 861 |
| 2 (2D): *numNoMatches*, *numNoInputs* | 239 |
| 3 (2D): *percNoInputs*, *percResults* | 29 |

**Table 4**
Average value of the interaction parameters in each cluster.

| Parameters | Cluster 0 | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|---|
| numExchanges | 18.10 | 9.76 | 9.69 | 30.14 |
| percConfirm | 0.62 | 0.59 | 0.57 | 0.66 |
| numNoMatches | 3.61 | 0.40 | 0.00 | 10.52 |
| numNoInputs | 0.00 | 0.00 | 0.00 | 0.04 |
| meanAVPs | 1.03 | 1.00 | 0.99 | 1.04 |
| successDial | 1.00 | 0.97 | 0.46 | 1.00 |
| sysActPerUserAct | 1.76 | 2.12 | 2.33 | 1.53 |
| meanReasks | 1.28 | 1.09 | 1.15 | 1.58 |
| maxReasks | 2.78 | 1.56 | 1.69 | 4.24 |
| percNoMatches | 0.18 | 0.03 | 0.00 | 0.03 |
| percNoInputs | 0.00 | 0.00 | 0.00 | 0.01 |
| Dialog acts | Cluster0 | Cluster1 | Cluster2 | Cluster3 |
| numFormal | 1.19 | 1.12 | 1.08 | 1.03 |
| numResults | 1.81 | 1.43 | 1.13 | 1.59 |
| numQueries | 13.78 | 7.99 | 6.97 | 22.41 |
| numStatusReports | 8.39 | 5.15 | 5.22 | 10.73 |
| numError | 1.02 | 0.00 | 1.06 | 3.24 |
| numInstructions | 3.39 | 2.31 | 2.09 | 4.17 |
| percFormal | 0.08 | 0.13 | 0.19 | 0.05 |
| percResults | 0.10 | 0.15 | 0.07 | 0.05 |
| percQueries | 0.78 | 0.82 | 0.62 | 0.76 |
| oercStatusReports | 0.46 | 0.53 | 0.42 | 0.04 |
| percError | 0.06 | 0.00 | 0.26 | 0.10 |
| percInstructions | 0.20 | 0.25 | 0.27 | 0.15 |

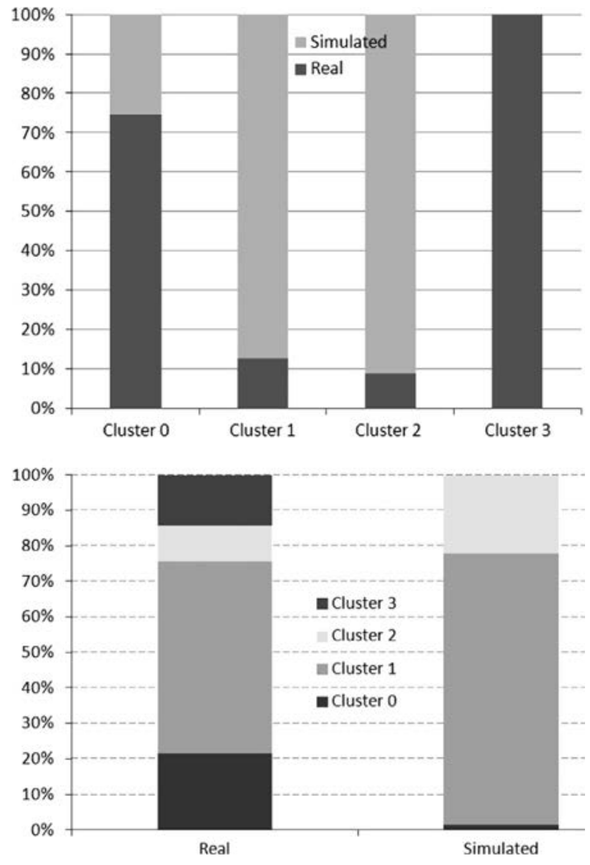

**Fig. 6.** Distribution of real and simulated dialogs among clusters.

the 82.21% of the dialogs. However, as shown in Fig. 6 (up), clusters 1 and 2 are 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 simu lated 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 behaviours 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.

### 4.2. *Evaluation of the complete proposed framework.*

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 meas ures 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.

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 previously 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.

**Table 5**
Evaluation results obtained with the DMs developed for the Let's Go system.

| DM | *Matching* (%) | *Coherence* (%) | *Error* (%) |
|---|---|---|---|
| $DM_1$ | 93.28 | 94.52 | 5.48 |
| $DM_2$ | 81.22 | 97.14 | 2.86 |

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

| System dialog acts | $DM_1$ | $DM_2$ |
|---|---|---|
| S_Confirm | 39.16 | 36.19 |
| S_Request | 20.08 | 18.21 |
| S_Inform | 40.39 | 45.37 |
| S_Other | 0.37 | 0.23 |

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

| System actions | $DM_1$ | $DM_2$ |
|---|---|---|
| Goal-directed actions | 73.16 | 78.89 |
| Grounding actions | 26.84 | 21.11 |

Moreover, the codification developed to represent the state of the dialog and the good operation of the MLP classifier make it possible for the number of responses that cause the failure of the system to be only 2.86% 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.

## 5. Conclusions

In this paper, we have combined different aspects from the areas of knowledge representation, natural language processing, user modeling and intelligent information retrieval to facilitate a personalized and more natural access to information and services by means of speech interaction. In order to do this, we have contributed a framework which can be used to develop adaptive spoken dialog systems.

Our proposal is based on the definition of a statistical metho dology for user modeling that anticipates the next user turn during the dialog and makes it possible to adapt the system dynamically to the 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 (predicted 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.

This methodology can be used to model slot filling tasks, which cover the most common application domains in which current dialog systems are involved, and for which defining a good dialog strategy can be difficult [38]. We have shown that our approach is scalable and can help to reduce the dimensionality of complex slot filling domains with a high number of 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 generated by a module that controls the application, for instance, to validate specific restrictions, apply privacy policies or carry out computations that define the next system response based on the application context. The output of such module can be considered transparently in our approach as an additional feature for the selection of the next system action.

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 considered, 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 satisfaction measures that complement the statistical measures employed.

## Acknowledgments

Work partially supported by Projects MINECO TEC2012 37832 C02 01, CICYT TEC2011 28626 C02 02, CAM CONTEXTS (S2009/TIC 1485).

## References

[1] E. Corchado, M. Graña, M. Wozniak, New trends and applications on hybrid artificial intelligence systems, Neurocomputing 75 (1) (2012) 61–63.
[2] M. McTear, Z. Callejas, Voice Application Development for Android, Packt Publishing, Birmingham, UK, 2013.
[3] R. Pieraccini, The Voice in the Machine: Building Computers that Understand Speech, MIT Press, Cambridge, USA, 2012.
[4] R. López-Cózar, M. Araki, Spoken, Multilingual and Multimodal Dialogue Systems, John Wiley & Sons Publishers, Hoboken, USA, 2005.
[5] M.F. McTear, Spoken Dialogue Technology: Towards the Conversational User Interface, Springer, Berlin, Germany, 2004.
[6] A. Tsilfidis, I. Mporas, J. Mourjopoulos, N. Fakotakis, Automatic speech recognition performance in different room acoustic environments with and without dereverberation preprocessing, Comput. Speech Lang. 27 (1) (2013) 380–395.
[7] W.-L. Wu, R.-Z. Lu, J.-Y. Duan, H. Liu, F. Gao, Y.-Q. Chen, Spoken language understanding using weakly supervised learning, Comput. Speech Lang. 24 (2) (2010) 358–382.
[8] D. Griol, L. Hurtado, E. Segarra, E. Sanchis, A statistical approach to spoken dialog systems design and evaluation, Speech Commun. 50 (8–9) (2008) 666–682.
[9] O. Lemon, Learning what to say and how to say it: joint optimisation of spoken dialogue management and natural language generation, Comput. Speech Lang. 25 (2011) 210–221.
[10] T. Dutoit, An Introduction to Text-to-speech Synthesis, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1996.
[11] B. Vesnicer, J. Zibert, S. Dobrisek, N. Pavesic, F. Mihelic, A voice-driven web browser for blind people, in: Proceedings of the Interspeech, 2003, pp. 1301–1304.
[12] J. Polifroni, G. Chungand, S. Seneff, Towards the automatic generation of mixed-initiative dialogue systems from web content, in: Proceedings of the Eurospeech, Geneva, Switzerland, 2003, pp. 193–196.
[13] S. D'Mello, A. Olney, C. Williams, P. Hays, Gaze tutor: a gaze-reactive intelligent tutoring system, Int. J. Human–Comput. Stud. 70 (5) (2012) 377–398.
[14] F. Metze, X. Anguera, E. Barnard, M. Davel, G. Gravier, Language independent search in MediaEval's Spoken Web Search task, Comput. Speech Lang. 28 (5) (2014) 1066–1082.
[15] M. Tsai, The VoiceXML dialog system for the e-commerce ordering service, in: Proceedings of the CSCWD, 2005, pp. 95–100.
[16] A. Stent, S. Stenchikova, M. Marge, Reinforcement learning of dialogue strategies with hierarchical abstract machines, in: Proceedings of the SLT, 2006, pp. 210–213.
[17] J. Chai, V. Horvath, N. Nicolov, M. Stys, N. Kambhatla, W. Zadrozny, P. Melville, Natural language assistant: a dialog system for online product recommendation, AI Mag. 23 (2002) 63–75.
[18] K. Kopp, M. Britt, K. Millis, A. Graesser, Improving the efficiency of dialogue in tutoring, Learn. Instr. 22 (5) (2012) 320–330.
[19] D. Litman, S. Silliman, ITSPOKE: an intelligent tutoring spoken dialogue system, in: Proceedings of the HLT/NAACL, 2004, pp. 233–236.
[20] H. Hofmann, A. Silberstein, U. Ehrlich, A. Berton, C. Muller, A. Mahr, Development of speech-based in-car HMI concepts for information exchange Internet apps, in: Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice, Springer Science+Business Media, 2014, pp. 15–28.
[21] J. He, A. Chaparro, B. Nguyen, R. Burge, J. Crandall, B. Chaparro, R. Ni, S. Cao, Texting while driving: is speech-based texting less risky than handheld texting? in: Proceedings of the Automotive'UI 13, 2013, pp. 124–130.
[22] G. Skantze, A. Hjalmarsson, C. Oertel, Turn-taking, feedback and joint attention in situated human-robot interaction, Speech Commun. 65 (2014) 50–66.
[23] W. Minker, T. Heinroth, P. Strauss, D. Zaykovskiy, Spoken dialogue systems for intelligent environments, in: Human-Centric Interfaces for Ambient Intelligence, Elsevier, Amsterdam, the Netherlands, 2010, pp. 453–478.
[24] T. Bickmore, K. Puskar, E. Schlenk, L. Pfeifer, S. Sereika, Maintaining reality: relational agents for antipsychotic medication adherence, Interact. Comput. 22 (2010) 276–288.
[25] O. Saz, S.-C. Yin, E. Lleida, R. Rose, C. Vaquero, W.-R. Rodríguez, Tools and technologies for computer-aided speech and language therapy, Speech Commun. 51 (10) (2009) 948–967.
[26] O. Horchak, J.-C. Giger, M. Cabral, G. Pochwatko, From demonstration to theory in embodied language comprehension: a review, Cognit. Syst. Res. 29–30 (2014) 66–85.
[27] C. Qu, W. Brinkman, Y. Ling, P. Wiggers, I. Heynderickx, Conversations with a virtual human: synthetic emotions and human responses, Comput. Human Behav. 34 (2014) 58–68.
[28] D. Griol, M. García-Jiménez, Development of interactive virtual voice portals to provide municipal information, Adv. Soft Comput. 151 (2012) 161–172.
[29] T. Shibata, Y. Egashira, S. Kurohashi, Chat-like conversational system based on selection of reply generating module with reinforcement learning, in: Proceedings of the IWSDS, 2014, pp. 227–231.
[30] F. Bessho, T. Harada, Y. Kuniyoshi, Dialog system using real-time crowdsourcing and twitter large-scale corpus, in: Proceedings of the SIGDIAL, 2012, pp. 227–231.
[31] F. Burkhardt, J. Zhou, “Askwiki': shallow semantic processing to query Wikipedia, in: Proceedings of the EUSIPCO, 2012, pp. 350–354.
[32] B. Hasler, P. Tuchman, D. Friedman, Virtual research assistants: replacing human interviewers by automated avatars in virtual worlds, Comput. Human Behav. 29 (4) (2013) 1608–1616.
[33] D. Griol, J. Molina, Z. Callejas, An approach to develop intelligent learning environments by means of immersive virtual worlds, J. Ambient Intell. Smart Environ. 6 (2) (2014) 237–255.
[34] T. Paek, R. Pieraccini, Automating spoken dialogue management design using machine learning: an industry perspective, Speech Commun. 50 (89) (2008) 716–729.
[35] J. Rouillard, Web services and speech-based applications around VoiceXML, J. Netw. 2 (1) (2007) 27–35.
[36] S. Seneff, M. Adler, J. Glass, B. Sherry, T. Hazen, C. Wang, T. Wu, Exploiting context information in spoken dialogue interaction with mobile devices, in: Proceedings of the IMUx, 2007, pp. 1–11.

[37] S. Kartakis, A. Design-and-Play, Approach to accessible user interface development in ambient intelligence environments, J. Comput. Ind. 61 (4) (2010) 318–328.
[38] S. Young, The Statistical Approach to the Design of Spoken Dialogue Systems, Technical Report, Cambridge University Engineering Department (UK), 2002.
[39] J. Schatzmann, K. Weilhammer, M. Stuttle, S. Young, A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies, Knowl. Eng. Rev. 21 (2) (2006) 97–126.
[40] D. Pierrakos, G. Paliouras, C. Papatheodorou, C. Spyropoulos, Web usage mining as a tool for personalization: a survey, User Model. User-Adapted Interact. 13 (4) (2003) 311–372.
[41] M. Eirinaki, M. Vazirgiannis, Web mining for web personalization, ACM Trans. Internet Technol. 3 (1) (2003) 1–27.
[42] J. Searle, Speech acts, An Essay on the Philosophy of Language, Cambridge University Press, Cambridge, UK, 1969.
[43] D. Traum, Speech acts for dialogue agents, Foundations of Rational Agency, Kluwer, Dordrecht, the Netherlands (1999) 169–201.
[44] H. Bunt, J. Alexandersson, J. Carletta, J. Choe, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, D. Traum, Towards an ISO standard for dialogue act annotation, in: Proceedings of the LREC, 2010, pp. 2548–2555.
[45] E. Levin, R. Pieraccini, W. Eckert, A stochastic model of human–machine interaction for learning dialog strategies, IEEE Trans. Speech Audio Process. 8 (1) (2000) 11–23.
[46] K. Scheffler, S. Young, Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning, in: Proceedings of the HLT, 2001, pp. 12–18.
[47] O. Pietquin, A framework for unsupervised learning of dialogue strategies (Ph.D. thesis), Faculte Polytechnique de Mons, 2004.
[48] K. Georgila, J. Henderson, O. Lemon, Learning user simulations for information state update dialogue systems, in: Proceedings of the Eurospeech, 2005, pp. 893–896.
[49] H. Cuayáhuitl, S. Renals, O. Lemon, H. Shimodaira, Human–computer dialogue simulation using hidden Markov models, in: Proceedings of the ASRU, 2005, pp. 290–295.
[50] M. Araki, T. Watanabe, S. Doshita, Evaluating dialogue strategies for recovering from misunderstandings, in: Proceedings of the IJCAI Workshop on Collaboration Cooperation and Conflict in Dialogue Systems, 1997, pp. 13–18.
[51] T. Watanabe, M. Araki, S. Doshita, Evaluating dialogue strategies under communication errors using computer-to-computer simulation, IEICE Trans. Inf. Syst. E81-D (9) (1998) 1025–1033.
[52] R. López-Cózar, A. de la Torre, J. Segura, A. Rubio, Assessment of dialogue systems by means of a new simulation technique, Speech Commun. 40 (2003) 387–407.
[53] E. Filisko, S. Seneff, Developing city name acquisition strategies in spoken dialogue systems via user simulation, in: Proceedings of the SIGdial, 2005, pp. 144–155.
[54] W. Eckert, E. Levin, R. Pieraccini, User modeling for spoken dialogue system evaluation, in: Proceedings of the ASRU, 1997, pp. 80–87.
[55] O. Pietquin, T. Dutoit, A probabilistic framework for dialog simulation and optimal strategy learning, IEEE Trans. Speech Audio Process. 14 (2005) 589–599.
[56] J. Bos, E. Klein, O. Lemon, T. Oka, DIPPER: description and formalisation of an information-state update dialogue system architecture, in: Proceedings of the SIGdial, 2003, pp. 115–124.
[57] S. Jung, C. Lee, K. Kim, D. Lee, G. Lee, Hybrid user intention modeling to diversify dialog simulations, Comput. Speech Lang. 25 (2) (2011) 307–326.
[58] R. Higashinaka, K. Sudoh, M. Nakano, Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems, Speech Commun. 48 (2006) 417–436.
[59] C. Seon, H. Kim, J. Seo, A statistical prediction model of speakers intentions using multi-level features in a goal-oriented dialog system, Pattern Recognit. Lett. 33 (2012) 1397–1404.
[60] F. Wang, K. Swegles, Modeling user behavior online for disambiguating user input in a spoken dialogue system, Speech Commun. 55 (2013) 84–98.
[61] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, S. Young, Agenda-based user simulation for bootstrapping a POMDP dialogue system, in: Proceedings of the HLT/NAACL, 2007, pp. 149–152.
[62] Y. Wilks, R. Catizone, S. Worgan, M. Turunen, Some background on dialogue management and conversational speech for dialogue systems, Comput. Speech Lang. 25 (2) (2011) 128–139.
[63] M. Ahmed, R. Riyaz, S. Afzal, A comparative study of various approaches for dialogue management, Int. J. Adv. Comput. Technol. 2 (4) (2013) 89–96.
[64] Y. Wilks, R. Catizone, S. Worgan, M. Turunen, Some background on dialogue management and conversational speech for dialogue systems, Comput. Speech Lang. 25 (2011) 128–139.
[65] C. Lee, S. Jung, K. Kim, D. Lee, G. Lee, Recent approaches to dialog management for spoken dialog systems, J. Comput. Sci. Eng. 4 (1) (2010) 1–22.
[66] S. Singh, M. Kearns, D. Litman, M. Walker, Reinforcement learning for spoken dialogue systems, in: Proceedings of the NIPS, 1999, pp. 956–962.
[67] S. Young, J. Schatzmann, K. Weilhammer, H. Ye, The hidden information state approach to dialogue management, in: Proceedings of the ICASSP, 2007, pp. 149–152.
[68] N. Roy, J. Pineau, S. Thrun, Spoken dialogue management using probabilistic reasoning, in: Proceedings of the ACL, 2000, pp. 93–100.
[69] S. Singh, D. Litman, M. Kearns, M. Walker, Optimizing dialogue management with reinforcement leaning: experiments with the NJFun system, J. Artif. Intell. 16 (2002) 105–133.
[70] P. Heeman, Combining reinforcement learning with information-state update rules, in: Proceedings of the HLT-NAACL, 2007, pp. 268–275.
[71] J. Williams, P. Poupart, S. Young, Partially observable Markov decision processes with continuous observations for dialogue management, Recent Trends in Discourse and Dialogue, Springer, Berlin, Germany (2006) 191–217.
[72] J. Williams, The best of both worlds: unifying conventional dialog systems and POMDPs, in: Proceedings of the Interspeech, 2008, pp. 1173–1176.
[73] P. Lison, Model-based Bayesian reinforcement learning for dialogue management, in: Proceedings of the Interspeech, 2013, pp. 457–461.
[74] S. Png, J. Pineau, B. Chaib-draa, Building adaptive dialogue systems via Bayes-adaptive POMDPs, IEEE J. Select. Top. Signal Process. 6 (8) (2012) 917–927.
[75] C. Hori, K. Ohtake, T. Misu, H. Kashioka, S. Nakamura, Recent advances in WFST-based dialog system, in: Proceedings of the Interspeech, 2009, pp. 268–271.
[76] L. Hurtado, J. Planells, E. Segarra, E. Sanchis, D. Griol, A stochastic finite-state transducer approach to spoken dialog management, in: Proceedings of the Interspeech, 2010, pp. 3002–3005.
[77] J. Planells, L. Hurtado, E. Sanchis, E. Segarra, An online generated transducer to increase dialog manager coverage, in: Proceedings of the Interspeech, 2012.
[78] T. Paek, E. Horvitz, Conversation as action under uncertainty, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000, pp. 455–464.
[79] H.H. Meng, C. Wai, R. Pieraccini, The use of belief networks for mixed-initiative dialog modeling, IEEE Trans. Speech Audio Process. 11 (6) (2003) 757–773.
[80] D. Griol, L.F. Hurtado, E. Segarra, E. Sanchis, Managing unseen situations in a stochastic dialog model, in: Proceedings of AAAI Workshop Statistical and Empirical Approaches for Spoken Dialogue Systems, Boston, USA, 2006, pp. 25–30.
[81] D. Griol, J. Iglesias, A. Ledezma, A. Sanchis, A dialog management methodology based on evolving fuzzy-rule-based (FRB) classifiers, in: Proceedings of the EAIS, Linz, Austria, 2014, pp. 1–8.
[82] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, PDP: Computational Models of Cognition and Perception, I, MIT Press, Cambridge, USA (1986) 319–362.
[83] S.M. Siniscalchi, T. Svendsen, C.H. Lee, An artificial neural network approach to automatic speech processing, Neurocomput. J. 140 (2014) 326–338.
[84] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.
[85] H. Hotta, M. Kittaka, M. Hagiwara, Word vectorization using relations among words for neural network, IEEJ Trans. Electron. Inf. Syst. 130 (1) (2010) 75–82.
[86] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text relatedness based on a word thesaurus, J. Artif. Intell. Res. 37 (1) (2010) 1–40.
[87] F. Zamora-Martínez, V. Frinken, S. Espana-Boquera, M. Castro-Bleda, A. Fischer, H. Bunke, Neural network language models for off-line handwriting recognition, Pattern Recognit. 47 (4) (2014) 1642–1652.
[88] M. Zissman, Comparison of four approaches to automatic language identification of telephone speech, IEEE Trans. Speech Audio Process. 4 (1) (1996) 31–44.
[89] I. Bazzi, J. Glass, Modeling out-of-vocabulary words for robust speech recognition, in: Proceedings of the ICSLP, 2000, pp. 401–404.
[90] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Word sense disambiguation with semantic networks, in: Lecture Notes in Computer Science, vol. 5246, 2008, pp. 219–226.
[91] H. Markert, U. Kaufmann, Z. Kara, G. Palm, Neural associative memories for the integration of language, vision and action in an autonomous agent, Neural Netw. 22 (2009) 134–143.
[92] L. Gajecki, Architectures of neural networks applied for LVCSR language modeling, Neurocomput. J. 133 (2014) 46–53.
[93] R. Socher, C. Chiung-Yu, A. Ng, C. Manning, Parsing natural scenes and natural language with recursive neural networks, in: Proceedings of the ICML, 2011, pp. 129–136.
[94] T. Sagara, M. Hagiwara, Natural language neural network and its application to question-answering system, Neurocomput. J. 142 (2014) 201–208.
[95] S. Bangalore, G. DiFabbrizio, A. Stent, Learning the structure of task-driven human–human dialogs, IEEE Trans. Audio Speech Lang. Process. 16 (7) (2008) 1249–1259.
[96] A. Berger, S. Pietra, V. Pietra, A maximum entropy approach to natural language processing, Comput. Linguist. 22 (1) (1996) 39–71.
[97] P. Haffner, Scaling large margin classifiers for spoken language understanding, Speech Commun. 48 (4) (2006) 239–261.
[98] F. Torres, L. Hurtado, F. García, E. Sanchis, E. Segarra, Error handling in a stochastic dialog system through confidence measures, Speech Commun. 45 (3) (2005) 211–229.
[99] A. Raux, B. Langner, A. Black, M. Eskenazi, Let's go public! taking a spoken dialog system to the real world, in: Proceedings of the Interspeech, 2005, pp. 885–888.
[100] A. Black, S. Burger, B. Langner, G. Parent, M. Eskenazi, Spoken dialog challenge 2010, in: Proceedings of the IEEE SLT, 2010, pp. 448–453.

[101] A. Schmitt, S. Ultes, W. Minker, A parameterized and annotated spoken dialog corpus of the CMU Let's Go bus information system, in: Proceedings of the LREC, 2012, pp. 3369–3375.
[102] J. Williams, I. Arizmendi, A. Conkie, Demonstration of AT&T Let's Go: a production-grade statistical spoken dialog system, in: Proceedings of the SLT, 2010, pp. 157–158.
[103] H. Hastie, N. Merigaud, X. Liu, O. Lemon, "Let's Go, DUDE!" Using the Spoken Dialogue Challenge to teach Spoken Dialogue development, in: Proceedings of the IEEE SLT, 2010, pp. 466–471.
[104] B. Thomson, K. Yu, S. Keizer, M. Gasic, F. Jurcicek, F. Mairesse, S. Young, Bayesian dialogue system for the Let's Go spoken dialogue challenge, in: Proceedings of the IEEE SLT, 2010, pp. 460–465.
[105] R. Vidal, Subspace clustering, IEEE Signal Process. Mag. 28 (2) (2011) 52–68.
[106] E. Muller, S. Gunnemann, I. Assent, T. Seidl, Evaluating clustering in subspace projections of high dimensional data, in: Proceedings of the VLDB, 2009, pp. 1270–1281.