

---

# ADAPTIVE DIALOGUE MANAGEMENT USING INTENT CLUSTERING AND FUZZY RULES

---

ACCEPTED (PEER-REVIEWED) VERSION

**David Griol, Zoraida Callejas**

Dept. of Software Engineering, University of Granada  
Periodista Daniel Saucedo Aranda s.n., 18071. Granada, Spain  
{dgriol, zoraida}@ugr.es

**José Manuel Molina, Araceli Sanchis**

Dept. of Computer Science,  
Universidad Carlos III de Madrid  
Avda. de la Universidad, 30, 28911. Leganés, Spain  
molina@ia.uc3m.es, masm@inf.uc3m.es

This is the peer reviewed version of the following article: [Griol, D, Callejas, Z, Molina, JM, Sanchis, A. Adaptive dialogue management using intent clustering and fuzzy rules. Expert Systems. 2020; e12630.](#) , which has been published in final form at <https://doi.org/10.1111/exsy.12630>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Use of Self-Archived Versions.

## ABSTRACT

Conversational systems have become an element of everyday life for billions of users who use speech-based interfaces to services, engage with personal digital assistants on smartphones, social media chatbots, or smart speakers. One of the most complex tasks in the development of these systems is to design the dialogue model, the logic that provided a user input selects the next answer. The dialogue model must also consider mechanisms to adapt the response of the system and the interaction style according to different groups and user profiles. Rule-based systems are difficult to adapt to phenomena that were not taken into consideration at design-time. However, many of the systems that are commercially available are based on rules, and so are the most widespread tools for the development of chatbots and speech interfaces. In this paper, we present a proposal to: i) automatically generate the dialogue rules from a dialogue corpus through the use of evolving algorithms, ii) adapt the rules according to the detected user intention. We have evaluated our proposal with several conversational systems of different application domains, from which our approach provided an efficient way for adapting a set of dialogue rules considering user utterance clusters.

**Keywords** Conversational Systems · Dialogue Management · Dialogue Rules, Evolving Classifiers · Clustering · User Modeling

## 1 Introduction

Conversational systems are computer programs that engage users in a dialogue using natural language [1, 2, 3]. The increasing maturity of speech and conversational technologies has made possible to integrate conversational interaction in a range of application domains, smart devices, and environments. Some good examples are personal assistants in mobile devices and smart speakers, educational tutoring agents, entertainment chatbots in open domains, online customer services that answer questions about products or services, question answering and triage systems, and robots that offer spoken communication [4, 5, 1, 6].

Spoken conversational systems usually require a sequence of interactions between the user and the system to gradually achieve the user's goals after several dialogue turns. A set of basic actions is repeated after each user utterance to recognize the sequence of words in the speech signal (Automatic Speech recognition, ASR), understand these words to obtain their meaning (i.e, extract and classify the information pieces that are useful for the domain of the system) (Spoken Language Understanding, SLU), consider the context features of the conversation and the results of the queries to the data repositories of the system to decide the next action/s for the system (Dialogue Management, DM), and generate a spoken message to provide a response to the user (Natural Language Generation and Text-to-Speech Synthesis, NLG and TTS) [7, 1, 3].

The *Dialogue Manager* (DM) is the component responsible for determining the next action to be taken by the system provided the user input. In order to do so, it must follow a dialogue strategy that can be computed using machine learning approaches or that may follow a predefined set of rules [1, 8, 9].

The design practices of conventional commercial dialogue systems are well established in industry and usually adopt a rule-based approach. This way, voice user interface (VUI) experts [10, 11, 12] handcraft a detailed dialogue plan based on their knowledge about the specific task and the business rules. In addition, designers commonly define the precise wording for the system prompts according to the dialogue state and context, and also the expected types of user's utterances for each turn.

As described in [13, 14], this approach is well-documented [15] and has been used to develop hundreds of successful commercial dialogue systems. Rule-based approaches are also an efficient alternative when the dialogue system must be available in an embedded device with important hardware constraints [16]. However, the resulting dialogue model lacks the flexibility to adapt the system according to new observations that were not considered at design time. Thus, new approaches must be considered in order to develop dialogue models that can be incorporate into state-of-the-art development environments and at the same time provide enough flexibility.

Most existing dialogue systems collect data to assess and improve their performance, including some form of quality assessment, time management representation, and business processing aspects. Dialogue systems can employ this information to provide personalized dialogue management strategies [17, 18, 19], for instance to adapt the services and information provided by the conversational system, optimize the overall time required to solve the users' queries and favour user engagement and fidelity.

While the combination of data with statistical methodologies has been applied to develop and improve different aspects of spoken dialogue systems (e.g., Hidden Markov models and Gaussian mixture models for Automatic Speech Recognition; Hidden Vector State model, Stochastic Finite State Transducers, Dynamic Bayesian Networks, Support Vector Machines and Conditional Random Fields for Spoken Language Understanding; Partially observable Markov Decision Processes and Bayesian Networks for Dialogue Management; Markov Decision Processes and Reinforcement Learning for Natural Language Generation) [1], they have not been sufficiently exploited to develop dialogue management models that can be easily adapted to different users, extension or the current application domain or new application domains. It is also required to have the explainability of the solution reached and the flexibility of transforming these statistical models into a set of rules to implement these models using already existing commercial infrastructures, such as Google DialogFlow, IBM Watson or Amazon Lex. These rules must be also designed to operate under partially observable settings.

This paper presents three main contributions:

- An approach to obtain a dialogue model for a specific task by means of a classification process using fuzzy-rule-based evolving classifiers. This allows obtaining a set of fuzzy rules that can be directly employed to develop a rule-based dialogue manager, making it possible to obtain new generation interfaces without the need for changing the already existing commercial infrastructures.
- An approach based on clustering to identify user intents, and how it can be incorporated to adapt the system rules to new observed user inputs by associating them to one of the clusters computed.
- To evaluate our proposal with conversational systems developed for domains with very different nature and complexity, different definitions for the semantics of the task, interaction languages, dialogue initiatives, confirmation and error handling and correction techniques, interaction, underlying technologies, and different techniques for acquiring the initial dialogue corpus (human-human, human-machine, Wizard of Oz technique).

The remainder of the paper is organized as follows. In Section 2 we describe the motivation of our proposal and review the related work in the areas of dialogue management and adaptation methodologies for conversational interfaces. Section 3 describes our proposal for automatically defining rule-based dialogue models and adapting them according to the user input clustering. Section 4.1 describes the dialogue systems used for evaluating our proposal. Section 5 presents the experimental set-up, the measures defined to evaluate the dialogue models and a discussion of the results obtained. Finally, Section 6 presents the conclusions and some guidelines for future work.

## 2 State of the art: user-adapted conversational interfaces

User adaptation in conversational systems can be achieved by means of the combination of techniques to capture and represent the information used to characterize and classify users, and flexible dialogue management strategies to adapt the dialogue model according to these information sources [1, 3].

### 2.1 Dialogue management methodologies

As described in the previous section, the design of the dialogue model is one of the most important tasks of a conversational system given that the selection of a specific system action depends on multiple factors (e.g., the outputs and confidence measures provided by the ASR and SLU modules; the channels and devices used for the interaction; the results of the queries to the data repositories; restrictions defined for the specific domain of the system; interaction initiatives directed by the user, system or both; confirmation strategies based on explicit or implicit confirmations, etc.). Due to these factors, the design of the dialogue model is at the core of conversational interface engineering and is largely responsible for user satisfaction. In addition, to consider these factors to decide the next system response, the dialogue manager needs to track the dialogue history and update its representation of the current state of the dialogue [1, 8].

Conversational systems designed for highly structured tasks and system-directed initiatives usually employ finite state-based dialogue models [20], for which the users' utterances are restricted to simple cases within the scope of the ASR and SLU modules. This approach is generally implemented using finite state automata with handcrafted rules.

Frame-based dialogue managers [21] use a slot-based frame structure to collect each information piece provided by the user. Users can fill more than one slot per dialogue turn following any order. Plan-based approaches [22] claim that the speaker's speech act is part of a plan and that it is the listener's task to identify and respond appropriately to this plan. Agent-based approaches can be employed when it is necessary to execute and monitor operations in application domains that change dynamically [23].

In all these approaches, application developers, together with voice user interface designers, typically handcraft DM strategies using rules and heuristics. However, there exist approaches to introduce variability in these systems. For

example, the emotional virtual agent PRIMER uses a rule-based dialogue manager [24] that is adaptive to the user as it does not only consider the current dialogue state, but also the emotional state of the conversation and user progress.

However, in certain scenarios it may be difficult and costly to foresee which form of system behavior will lead to a quick and successful completion of the dialogue. This has motivated the research community to find ways for automating dialogue learning using statistical models trained with real conversations [18, 25, 26, 27]. These models allow to explore a wider range of strategies to model the variability of user behaviours.

Statistical approaches to DM can be classified into three main categories: dialogue modeling based on reinforcement learning (RL), corpus-based statistical dialogue management, and example-based dialogue management. Example-based approaches can be considered a specific case of corpus-based statistical dialogue management, given that they predict the next system action when the dialogue manager finds dialogue examples that have a similar dialogue state to the current one [20].

An extended methodology for learning dialogue strategies models human-computer interaction as an optimization problem using Partially Observable Markov Decision Process (POMDPs) and reinforcement methods [28, 25]. This approach uses multiple hypotheses of the current dialogue state to explicitly represent uncertainty. However, it is limited to small-scale problems, since they require many hand-crafted features for the state and action space representation [29]. Other interesting approaches for statistical DM are based on modeling the system by means of Hidden Markov models (HMMs) [30], stochastic finite-state transducers [26], Bayesian networks [31], or recurrent neural networks [32] and Reinforcement Deep Learning [33].

In this paper, we propose a hybrid approach to dialogue management that seeks to combine the benefits of rule-based and statistical techniques in a single framework. To do this, we propose a statistical technique to automatically extract the set of rules for dialogue management from a labeled dialogue corpus. Several authors have followed a similar approximation, for example, using discriminative classification models to learn information state updates [34], probabilistic rules whose values are estimated from dialogue data using Bayesian inference [8], or procedural dialogue management methodologies based on defining dialogue trees [35].

Our proposal differs from the aforementioned approaches in several main aspects. Our proposal models the dialogue by means of a classification process based on fuzzy-rule-based evolving classifiers, which considers the complete history of the dialogue as input. Fuzzy logic is an efficient means to reason over the data and provide a varied system behaviour while maintain the explainability of the solution reached. Recently, [36] employed fuzzy rules to build a recommender system that follows a conversational logic in which the user is prompted about their preferences at each step. In their case, the fuzzy logic is aimed at optimizing the recommendation, while for us it is way of obtaining a set of rules that can be directly employed to develop a rule-based dialogue manager, making possible to obtain new generation interfaces without the need for changing the already existing commercial infrastructures.

## 2.2 Identifying user intents

As explained in the introduction section, the rules generated for the dialogue manager must be adaptable to different users. A recent review of the literature of the last decade about conversational agents in business [6], has highlighted as one the challenges for the future the definition and management of user profiles, the context of interaction, and the detection methods of user's intention in order to provide personalized responses.

Indeed, to provide a positive user experience, dialogue systems should ideally adapt to the behavior of individual users or particular user groups. User profiling and behaviour adaptation is also a hot topic in human-robot interaction, specially for the detection of social cues [37].

In order to be user aware, dialogue systems must identify user profiles and be able to tailor the interaction the them. In [38], the authors present three basic forms of showing user awareness in conversational agents: being aware of the previous interactions, re-prompt users when there is no input, reword and re-prompt messages with more detailed information based on previous interaction. These guidelines are valid for all system users and at the same type consider specific information about each individual. Another approach is to group users that behave similarly and generate adaptation categories.

Research in techniques for user modeling has a long history within the fields of conversational systems [9]. Most of the proposed user models are used to generate dialogue corpora by means of the interaction between the conversational system and the user model. Different types of statistical techniques have been proposed to achieve this goal: n-gram models[39], graph-based models [40], Hidden Markov Models [41], logistic regression [42], EM-based algorithms [43]. This kind of models have been used to automatically evaluate conversational systems (particularly at the earlier stages of development), determine the effects of changes to the system's functionalities or evaluate the capacity of the dialogue

manager to react to unexpected situations. However, these models are not used to define user profiles and adapt the dialogue manager.

A recent survey of intent detection methods for dialogue systems is presented in [7]. This is one of the main tasks for the NLU module in order to detect the dialogues acts in the user utterance and provide this information to the dialogue manager. A range of statistical techniques have been proposed to complete this task, from traditional techniques, such as Naive Bayes [44] or Support Vector Machines [45], to mainstream methods, such as Convolutional Neural Networks [46], word embeddings [47] Recurrent Neural Networks [48], Long Short-Term Memory Networks [49] or Gated Recurrent Units [50].

Clustering techniques have been proposed to avoid the time-consuming task required to manually analyze the set of user questions and create a taxonomy of intents to be assigned to the appropriate system responses [51]. These techniques have been employed to mine dialogue corpora with different purposes. For example, [52] used clustering to group different types of comments provided by annotators and use them to anticipate dialogue breakdown. [53] propose a clustering technique to extend the standard K-means algorithm to simultaneously cluster user and system utterances considering their adjacency information.

Clustering relies on distance measures between the user inputs being classified. Measuring semantic text similarity has been a key topic in Artificial Intelligence, Natural Language Processing, and information retrieval for many years [54]. Different techniques has been proposed for this task [55]. A set of techniques uses bag of words to represent documents using vectors and calculate distances in the vector space model using the cosine similarity. The second group of techniques are based on the assumption that two text sequences are semantically equivalent if their words or expressions can be aligned. The third group of approaches use machine learning approaches combining different measures, features and techniques: graph-based approaches [56], word embeddings [57], convolutional neural networks [58], dynamic finite state machines [59], etc.

[60] showed the feasibility of this approach with application domains with a varied number of slots and also with real and simulated users. [61] describes a modification of the K-Means algorithm adapted to question-answering websites and forums. [62] presents a proposal to use clustering to create directed graphs with the main transitions between topics among dialogues.

These techniques are also proposed in [63, 64] to identify the topics and sequences of topics during the dialogue. Different methods are then proposed to use this information to manage the dialogue (i.e., model the transitions between topics: Hidden Markov Models Paul12, Latent Dirichlet Allocation [65], Deep Learning techniques [66], POMDPs [67] or ad-hoc methods based on the definition of heuristics [68]. We have used clustering in previous work to generate user profiles that can be of interest to assess the quality of the system and/or predict user satisfaction [69].

The method that we propose to identify the user intents is based on measuring the semantic textual similarity between the user sentences available in a data corpus for the target domain. This information is then used with the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [70] to obtain an initial representation of the domain based on the distance matrix computed. The set of clusters obtained represents the users' sentences that can be assigned to the same system intent. After the system receives a new user input, it is assigned to one of the clusters and this information is taken into account in the conditions of the fuzzy dialogue rules for a fine grained decision of the next system response, setting the basis for an integrated adaptive dialogue manager development procedure that can be easily interpreted, adapted and implemented with commercial platforms.

### 3 Our proposal for defining user's adaptive dialogue models

Figure 1 shows the architecture of the proposed framework to design rule-based dialogue managers. This figure shows the training and run time operations of the proposal. In the training stage, the user utterances that are available in the training corpus are used to calculate the text similarities among them.

For the corpus in English we have computed the semantic textual similarity (STS) between each pair of user phrases in the corpus using the API of the University of Maryland [54]. This API uses an hybrid word similarity model based on Latent Semantic Analysis (LSA) Word Similarity hypothesis (words occurring in the same contexts tend to have similar meanings) based on a corpus with three billion words. The word co-occurrence models are combined with WordNet to boost LSA similarity and deal with words of many senses.

The previous measure is not available for Spanish, so for languages different to English we have used cosine similarity, a metric that is very widespread to compare documents. It measures the angle between two vectors projected in a multidimensional space. In our case, two arrays containing the word counts, where each dimension corresponds to a word in the user input. To perform the calculations we have implemented a script in Python using the library *scikit* to

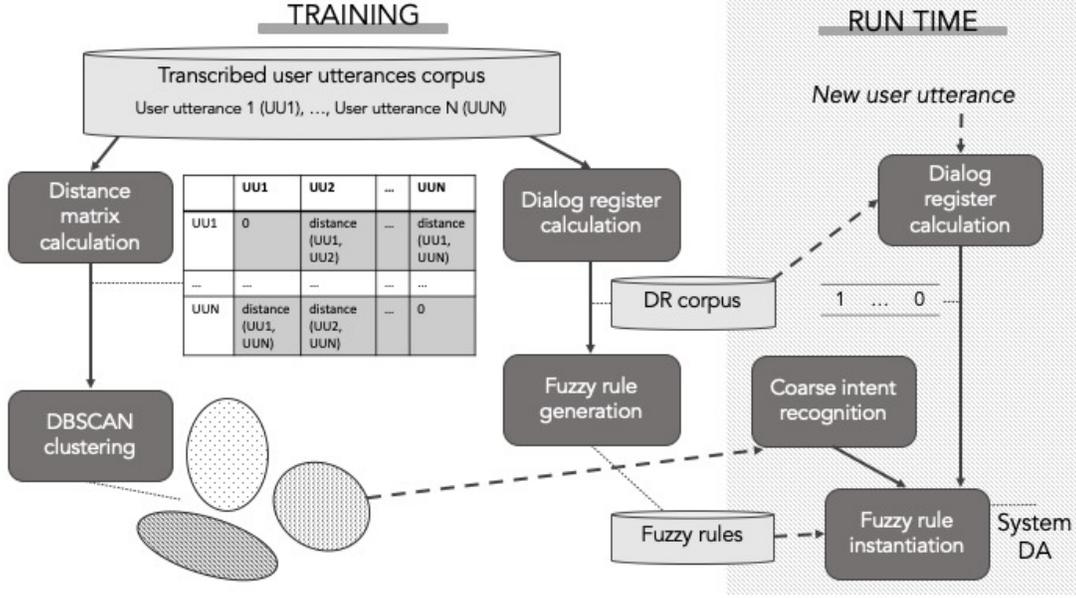


Figure 1: Framework designed for adaptive dialogue management using intent clustering and fuzzy rules

vectorize the phrases, *pandas* and *scipy* to generate data frames and the squared distance matrix, and *klearn* for the computation of cosine distance.

The distance matrix obtained in this step is used as input for a clustering algorithm. In order to be able to incorporate the clustering information in a system, the clustering algorithm must be flexible so that new user inputs are assigned to previously obtained clusters. As the most accurate number of clusters may differ in different application domains, we propose to use a method that does not require to specify a predetermined number of clusters.

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm uses the neighborhood size as parameter and the distance matrix as metric. It can be applied to very large sets of samples. Clusters can be any shape given that they are viewed as areas of high density (core samples) separated by areas of low density (non-core samples). A cluster is defined as a set of close core samples and non-core samples, which are close to a core sample. Two parameters (*min - samples* and *eps*) are used to determine the density required to form a cluster. We have used  $eps = 0.4$  and  $min - samples = 3$ , which after experiments with different schemes in the different corpora used, obtained the most balanced proportion of samples in each cluster and being robust enough to noise in all the application domains.

Our approach to integrate statistical dialogue management strategies in conversational interfaces extends our proposals in [18, 71]. This methodology is focused on slot-filling tasks, in which a slot-based structure is used to store the information provided by the user during the dialogue and this information and the dialogue model is considered to select the next system action. The training corpus is used to determine the number of slots (i.e. entities) to be considered for the task.

We represent dialogues as a sequence of pairs (*system - turn, user - turn*) at each time  $i$ , expressed in terms of dialogue acts:

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

The objective of the dialogue manager at each time  $i$  is to select the next system response based on the sequence of pairs until this moment:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\text{argmax}} P(A_i | (A_1, U_1) \dots (A_{i-1}, U_{i-1})) \quad (1)$$

To simplify the search for practical tasks, we define a data structure in order to establish a partition in the history of the dialogue preceding time  $i$ . This slot-based data structure, which we call *Dialogue Register (DR)*, contains the

information provided by the user throughout the previous history of the dialogue. Consequently, the selection of the best  $A_i$  is given by:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\text{argmax}} P(A_i | DR_{i-1}, A_{i-1}) \quad (2)$$

The selection of the next system response is then based on the task-dependent information pieces stored in the dialogue register and the system response selected in the previous turn. The specific values provided by the user are important for accessing databases and for constructing the output sentences of the system. However, these information pieces are coded in the  $DR$  according three possible values:

- (0): The value for the slot has not been provided;
- (1) The value is known with a confidence score that is higher than a given threshold;
- (2): The value of the slot has a confidence score that is lower than the given threshold.

We have used the  $DR$  abstraction successfully in our previous work to solve Equation 2 by means of a classification process that considers the dialogue history until the current moment of the dialogue (represented by means of the data pieces stored at the  $DR$  and the last system response). We propose the use of a fuzzy-rule-based evolving classifiers to discover the set of rules that are implicit in a dialogue corpus. Specifically, we propose the use of the  $eClass$  (evolving Classifier) family. During the training of these classifiers, a set of fuzzy rules that describes the most important observed features for the classification of each class (i.e., system prompt) is formed. These rules can also be constantly adjusted to the available training data Angelov04TSM.

The  $eClass0$  model consists of several fuzzy rules per class (the number of rules depends on the heterogeneity of the input data of the same class). A fuzzy rule in the  $eClass0$  model has the following structure:

Rule $_i = IF(Feature_1 \text{ is } P_1) \text{ AND } \dots$   
 $\dots \text{ AND } (Feature_n \text{ is } P_n) \text{ AND } (Cluster \text{ is } C_m)$   
 THEN Class =  $c_i$

where  $i$  represents the number of the rule;  $n$  is the number of input features (number of entities corresponding to the different slots defined for the  $DR$ ); the vector  $Feature$  stores the observed features, whereas vector  $P$  stores the values of the features of one of the prototypes (coded in terms of the tree possible values,  $\{0, 1, 2\}$ ). The estimation of the cluster in which the user utterance is classified taking the word similarities into account ( $C_m$ ) is considered as an additional feature in the conditions of the rules to select the next system response. Each class  $c_i \in \{\text{set of different classes}\}$  is then associated to a specific system action (response).

A specific system action can be represented by several rules, depending on the heterogeneity of the samples that represent the same action. Thus, a class could be represented by one or several prototypes (i.e., data sample that groups several samples which represent a specific system action). The different prototypes that represent a system action are obtained from the input data and they are updated constantly. The  $eClass0$  evolving classifier is trained by means of the set of steps that are described in [72]:

1. Classify each new sample in a group represented by a prototype To do this, the sample is compared with all the prototypes previously created.
2. Calculate the potential (Cauchy function of the sum of distances between a certain data sample and all other data samples in the feature space) of the new data sample to be a prototype. Based on the potential of the new data sample to become a prototype, it could form a new prototype or replace an existing one. The potential can be calculated using the euclidean or the cosine distance angelov08Evolving.
3. All the existing prototypes are updated considering the new data sample. A new prototype is created if its value is higher than any other existing prototype. Existing prototypes could also be removed.

## 4 Practical application domains considered for the empirical evaluation of the proposal

We have employed three dialogue systems for evaluating our proposal: Let's Go, Dihana, and EDECAN. Table 1 summarizes the main characteristics of the systems and corresponding corpora available for each of them.

### 4.1 Let's Go system

Let's Go is a spoken dialogue system developed by the Carnegie Mellon University to provide bus schedule information in Pittsburgh. The system has had many users since it was made available for the general public in 2005 - 20,000 calls

	Let's Go	Dihana	EDECAN
Initiative	Mixed and system directed	Mixed	Mixed
Error Handling	Yes (Explicit confirmations)	Yes (Explicit, Implicit and Mixed confirmations)	Yes (Implicit and Explicit confirmations)
Initial Corpus Collection	Human-computer	WOz Technique	Human-Human and supervised WOz Technique
Initial DM	Rule-based	Rule-based	Mixed (statistical model and rules)
Domains	Information provision	Information provision	Information Provision and Booking Operations
Languages	English	Spanish	Spanish, Catalan and Basque
Number of user dialogue acts	16	36	13
Number of system dialogue acts	36	51	21
Number of dialogues	338	900	300

Table 1: Main characteristics of the tasks and corpora used in the evaluation

collected just from March to December of 2005 [73]. The information provided by the system covers a subset of 5 routes and 559 bus stops.

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 dialogue system that provided its service to real users. This poses a challenge to build realistic user models and find new dialogue 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 dialogue 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.

Figure 2 shows an example of a dialogue extracted from the Let's Go corpus. 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 dialogue acts that has been classified into 16 categories. 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 turn in a dialog.

```

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!

```

Figure 2: Example of dialogue extracted from the Let's Go corpus

A total of 36 system dialogue acts were defined. These dialogue acts can also be classified into 5 groups: *formal* (dialogue 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). The Let's Go task considers up to 455 different values for the task-dependent user dialogue 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 dialogue manager more efficient and scalable, and suitable for complex application domains.

Figure 3 shows an example of the labeling of a user and system dialogue turns for the Let's Go task.

<b>User turn:</b> I want to leave from Downtown. <i>0.59 inform(from.desc=downtown)</i>
<b>System turn:</b> East Pittsburgh Bus Schedules. Say a bus route, like 28X, or say I'm not sure. <i>hello(), request(route), example(route=28x), example(route=dont_know)</i>

Figure 3: An example of the user and system dialogue acts defined for the Let's Go corpus

## 4.2 Dihana

Dihana [27] is a dialogue system developed to provide train scheduling information related to routes in Spain. A corpus of 900 dialogues (10.8 hours) was acquired for the task by means of the Wizard of Oz (WoZ) technique with 225 real users, for which an initial dialogue strategy was defined by experts.

A set of 20 scenarios was used to carry out the acquisition. Each scenario defined one or two objectives to be completed by the user and the set of attributes that they must provide. The corpus consists of 6,280 user turns and 9,133 system turns, with an average number of 7.6 words per turn and the vocabulary has a size of 811 words.

Users can ask about 5 different intents (timetables, fares, train types, and duration of a trip) and provide 10 entities (origin city, destination city, departure date, arrival date, departure hour, arrival hour, ticket-class, train type, order number, and services list). The *DR* defined for the Dihana task consists of these 15 elements. Users can also provide the task-independent (*Acceptance*, *Rejection* and *(Not-Understood* intents).

A total of 51 system responses were defined for the task (classified into confirmations of concepts and attributes, questions to require data from the user, and answers obtained after a query to the database).

Figure 4 shows an example of a dialogue in the Dihana corpus.

S1: Welcome to the railway information system. How can I help you? U1: I want to know timetables to Barcelona. S2: Tell me the departure date. U2: Tomorrow. S3: Do you want to leave tomorrow? U3: Yes. S4: There are several trains. The first one leaves at 08:54 and the last one at 23:45. Anything else? U4: Tell me the duration of the trip for the first train. S5: The duration of the trip for the train leaving at 08:54 is 1 hour and 43 minutes. Anything else? U5: Yes, tell me the price of a ticket in business class. S6: The price in business class is 54.50 euros. Anything else? U6: No, thank you. S7: Thank you for using the railway information system. Goodbye.
--

Figure 4: Example of a dialogue acquired for the Dihana dialogue task (translation from Spanish to English)

Figure 5 shows an example of a labeling of a user and system dialogue turns for the Dihana corpus.

<b>User turn:</b> I want to know timetables for trains leaving from Valencia to Madrid. <i>(Timetables) [0.95]</i> <i>Origin: Valencia [0.54]</i> <i>Destination: Madrid [0.76]</i>
<b>System turn:</b> The price of a business ticket for the AVE train leaving at 18.35 is 53.40 euros. <i>(Fares)</i> <i>Ticket-class: Business</i> <i>Type-train: AVE</i> <i>Departure-hour: 18.35</i> <i>Price: 53.40</i>

Figure 5: An example of the user and system dialogue acts defined for the Dihana corpus

## 4.3 EDECAN

EDECAN [74] is a multilingual, multimodal, mixed-initiative information system for booking sport facilities. Users can book, cancel and ask for the availability of sport facilities. Four different intents (*Availability*, *Booking*, *Booked*, *Cancellation*) and six entities (*Sport*, *Hour*, *Date*, *Court-Type*, *Court-Number*, and *Order-Number*) were defined to label the user utterances. The *DR* defined for the EdecAN task consists of these 10 elements. Users can also provide the task-independent (*Acceptance*, *Rejection* and *(Not-Understood* intents).

The system utterances were annotated considering a set of 15 task-dependent intents to inform users about the result of a request (sport facilities that are available, a new booking of sport facilities, sports facilities already booked, and

cancellation of sport facilities), to ask the users for values required to complete a query (sport, date, hour, and court-type), confirm the values provided, inform about violation of rules and regulations defined for the sport service, and request to select one of the available facilities. Six task-independent intents were also defined: Opening, Closing, Non-Understood, New-Query, Acceptance and Rejection.

A set of 300 dialogues recorded with 24 users was labeled in terms of the previously described user and system dialogue acts. The corpus consists of 1511 user turns with an average words per user turn of 6.83. The languages involved in the acquisition were Spanish and Catalan. The dialogues in the corpus have different levels of complexity depending on the number of queries that are completed. Figure 6 shows an example of a dialogue for the Edecan task.

<p>S1: Welcome to the sport service. How can I help you?          U1: I want to book a basketball court.          S2: You can see the available courts on the screen.          U2: I want a court for November 12th in the afternoon.          S3: Sorry, do you want a court from 15 to 21 hours?          U3: Yes.          S4: There are several courts available on Friday November 12th. Please select one of them.          U4: The one at 7pm.          S5: There is only one court with the required features, do you want to book it?          U5: Yes.          S6: The basketball court has been booked. Anything else?          U6: Yes, I want to know my list of booking courts.          S7: You can see the courts that you have booked on the screen. Anything else?          U7: Yes, cancel the one at 12 am.          S8: The specified court has been released. Anything else?          U8: No, thank you.          S9: Thank you for using the sport service. Goodbye.</p>
--

Figure 6: Examples of a dialogue acquired for the EDECAN dialogue task (translation from Spanish to English)

Figure 7 shows an example of a labeling of a user and system dialogue turns for the EDECAN corpus.

<p><b>User turn:</b> I want to book a padel court for 25th May at 9'clock in the morning.  <i>(Booking)</i> [0.89]  <i>Sport:</i> Padel [0.96]  <i>Date:</i> 05-25 [0.12]  <i>Hour:</i> 09.00-10.00 [0.85]</p>
<p><b>System turn:</b> Do you want to play on the 25th May?  <i>(Confirmation-Date)</i></p>

Figure 7: An example of the user and system dialogue acts defined for the EDECAN corpus

## 5 Evaluation

We have used the three corpora described in the previous section to apply the text similarity measurement and clustering approaches described in Section 3. Table 2 shows the set of clusters detected for each one of the tasks using our proposal. A total of seven clusters have been defined for the Let's Go task, seven for the Dihana Task, and eight clusters for the Edecan task.

A 5-fold cross-validation process has been used to carry out the evaluation. The initial corpus for each dialogue task was randomly split into five subsets (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 dialogue models. A validation subset (20%) was extracted from each training set.

We have assessed the appropriateness of the dialogue manager decisions whether considering or not the result of the user groups decided by the clustering approach. To do this, two dialogue managers have been generated using the training corpus for each one of the tasks: the dialogue model  $DM_1$  was generated learning the fuzzy rules without using the clustering approach to predict the user's intention and the dialogue model  $DM_2$  considers the complete framework described in Figure 1 for selecting the next system response.

From our previous work on statistical dialogue management [27], we propose three measures to evaluate the quality of the responses selected by the statistical dialogue manager. These measures are calculated by comparing the answer automatically generated by the dialogue models with regard to the reference answer annotated in the test corpus for each sample. This way, the evaluation is carried out turn by turn. Thus, the aim with these measures is not to evaluate the complete dialogue as a unit, but to assess the appropriateness of the dialogue 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)

Table 2: Clusters obtained for each one of the dialogue tasks

Let's Go		
Cluster	Explanation	Examples of sentences
Cluster_1	Reset dialogue	<i>Start again., Restart.</i>
Cluster_2	Ask for help	<i>Help</i>
Cluster_3	Provide confirmation	<i>Yes. Correct. No, to Duquesne</i>
Cluster_4	Ask for time	<i>Around 4.15pm. At 16.50</i>
Cluster_5	Provide arrival	<i>McKeesport, To Allegheny Sq. West</i>
Cluster_6	Provide departure	<i>From South Busway</i>
Cluster_7	Answer for new query	<i>New query</i>
Dihana		
Cluster	Explanation	Examples of sentences
Cluster_1	Query providing entities	<i>I want timetables from Granada to Valencia next Monday. Tell me the price of the last train.</i>
Cluster_2	Provide confirmation	<i>Yes, That's correct.</i>
Cluster_3	Ask for timetables not providing entities	<i>Timetables information., I want to know timetables.</i>
Cluster_4	Provide city	<i>I want to go to Madrid, Barcelona</i>
Cluster_5	Provide date	<i>8th December, Next Tuesday</i>
Cluster_6	Ask for timetables and fares	<i>I want to know timetables and fares. Timetables and prices to Valencia.</i>
Cluster_7	Provide type of train	<i>I want to travel in an AVE train. The fastest one.</i>
Edecán		
Cluster	Explanation	Examples of sentences
Cluster_1	Ask for a query	<i>I want to book a paddle or tennis court on Thursday morning, Yes, can I book the court on Wednesday at 11?</i>
Cluster_2	Rejection and query	<i>No, to book a court.</i>
Cluster_3	Provide Confirmation	<i>Yes, Correct, Not</i>
Cluster_4	Provide type of court and hour	<i>A tennis court at 9. A basketball indoor court from 11 to 13.</i>
Cluster_5	Cancellation	<i>I want to cancel one of my bookings for Thursday. I want to cancel my booking.</i>
Cluster_6	Ask for current bookings	<i>I want to know the information about my booking courts</i>
Cluster_7	Provide date	<i>For the weekend. On Saturday</i>
Cluster_8	Provide hour	<i>8pm. On Wednesday morning</i>

Table 3: Evaluation results of the DMs developed for the three dialogue tasks

	Let's Go			Dihana			Edecán		
	Matching	Coherence	Error	Matching	Coherence	Error	Matching	Coherence	Error
$DM_1$	89.08%	93.07%	5.69%	88.46%	92.01%	6.76%	85.33%	89.26%	7.25%
$DM_2$	92.03%	96.15%	4.23%	91.06%	95.26%	5.25%	88.21%	91.18%	5.11%

*Matching*: the percentage of responses provided by the dialogue 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 dialogue manager that are coherent with the current state of the dialogue although they are not necessarily the same that the reference answer; iii) *Error*: the percentage of answers provided by the dialogue manager that would cause the failure of the dialogue. We have also evaluated the overall success rate of the dialogues (*Success*).

The measure *Matching* is automatically calculated, evaluating whether the responses provided by  $DM_1$  or  $DM_2$  are equal to the reference answers for the sample. 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 dialogue 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 3 shows the results of the proposed evaluation of the  $DM_1$  and  $DM_2$  models for each one of the three dialogue tasks. As it can be observed, the values obtained for the matching and coherence measures show that the  $DM_2$  dialogue manager deviates from the  $DM_1$  dialogue 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.

With regard the success rate, the  $DM_1$  and  $DM_2$  dialogue models for the Let's Go task respectively achieve a 89% and 92% percentages of successful dialogues. With regard a literature of different studies that have used the Let's Go dialogue corpus, the initial version of the system provided complete dialogues with a 79% success rate with an average length of a dialogue of 14 turns and a minimum number of 6 necessary confirmations [75]. The version of the system presented in [76] 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 [77], 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

Table 4: Percentages of system dialogue acts using  $DM_1$  and  $DM_2$ 

	Let's Go		Dihana		Edecan	
	$DM_1$	$DM_2$	$DM_1$	$DM_2$	$DM_1$	$DM_2$
S_Confirm	37.32	35.10	31.84	28.86	38.56	37.11
S_Request	21.12	20.36	27.86	29.06	19.47	22.18
S_Inform	41.21	43.81	39.57	41.40	41.45	40.34
S_Other	0.35	0.26	0.73	0.68	0.52	0.37

Table 5: Percentages of goal directed and grounding actions using  $DM_1$  and  $DM_2$ 

	Let's Go		Dihana		Edecan	
	$DM_1$	$DM_2$	$DM_1$	$DM_2$	$DM_1$	$DM_2$
Goal-directed actions	74.18	77.32	73.24	76.12	69.31	71.88
Grounding actions	25.82	22.68	26.76	23.88	30.69	28.12

Dialogue Challenge are compiled in [78]. The four systems described in this paper respectively provide success rates of  $64.8 \pm 5.0$ ,  $37.7 \pm 6.2$ ,  $89.3 \pm 3.6$ , and  $74.7 \pm 4.8$ .

The  $DM_1$  and  $DM_2$  dialogue models for the Dihana task respectively achieved a 85% and 87% percentages of successful dialogues. With regard to the previous studies that have used the Dihana corpus, a 72.60% percentage of successful dialogues were obtained for the statistical dialogue model and user simulation techniques proposed in [79], a percentage of 82.90% successful dialogues was obtained in a HMM-based model proposed in [80], a percentage of 83.64% successful dialogues was obtained with the statistical methodology proposed in [27].

The  $DM_1$  and  $DM_2$  dialogue models for the Edecan task respectively achieved a 83% and 80% of successful dialogues. With regard to the previous studies that have used the Edecan corpus, a 78.80% percentage of successful dialogues were obtained for the statistical dialogue model and user simulation techniques proposed in [18], a percentage of 73.40% successful dialogues was obtained in a stochastic finite-state transducers model proposed in [74].

The detailed study of the system responses provided by the  $DM_2$  dialogue models showed that by considering the information provided by the user intention clustering were 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 dialogue acts at each turn. Also, Let's Go and DIHANA were designed to be system-directed at specific moments of the interaction. Our results show that the dialogue manager learnt with our proposal was able to find alternative outputs that are coherent with those observed in the corpora, but are not restricted to the fixed choice of system-directed answers as in the original systems.

The proposed codification of the information in the Dialogue Register 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 dialogue (*Coherence* measure) and allows to achieve the objective(s) of the dialogue in a less number of turns. Moreover, the codification developed to represent the state of the dialogue 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 4.23%, 5.25% and 5.11% for the  $DM_2$  dialogue models, reducing the percentages obtained with the  $DM_1$  dialogue features.

With respect to the dialogue style features, we measured the balance between different types of system dialogue acts using  $DM_1$  and  $DM_2$ . The results, showed in Table 4, indicate that using the  $DM_2$  dialogue models 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 proposed dialogue model.

In addition, we grouped all user and system dialogue acts into "goal-directed" (actions to provide or request information) and "grounding" actions (dialogue formalities, unrecognized actions, confirmations, and negations). The results in Table 5 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  $DM_1$  dialogue model.

## 6 Conclusions and future work

Successful applications of expert systems suggest that it will have increasingly greater impact in the coming years for customer relationship management within the service industry. The development of personalized conversational

systems is key in the pathway towards this scenario. In this paper, we have presented an approach to create adapted dialogue models for these systems. Our proposal uses evolving classifiers to automatically obtain a set of fuzzy rules that can be directly employed to develop a rule-based dialogue manager, thus reducing the considerable effort and time that is required to manually define the dialogue strategy and allowing the easy implementation of the dialogue model by means of the current commercial infrastructures. A clustering approach is also proposed to estimate user intents by means of the text similarities among the user utterances. This information is used as a term in the conditions of the set of rules for selecting the next system response.

We have used three corpora of practical dialogue systems with different interaction languages, complexity and underlying technologies to evaluate the proposed framework. With respect to the comparative assessment of the proposed framework whether integrating the clustering approach with the intent intention or not, the results show that the number of coherent responses provided by the complete dialogue model increases with respect to not using the clustering approach, while the number of responses that lead to dialogue failure decreases. The proposed dialogue model also improves the confirmation and error correction rates for the different tasks. The results obtained outperform the results reported in the literature for all the tasks.

For future work we plan to apply the proposed technique to multi-domain tasks in order to measure the capability of our methodology to adapt efficiently to contexts that can vary dynamically. We also want to combine our proposal to facilitate the interaction using also additional input and output modalities. Finally, we also intend to extend the evaluation of the system considering additional measures related to user's profiles that complement the proposed adaptation.

## Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823907 (MENHIR project: <https://menhir-project.eu>), and the Spanish project TEC2017-88048-C2-2-R.

## References

- [1] Michael F. McTear, Zoraida Callejas, and David Griol. *The Conversational Interface. Talking to smart devices*. Springer, 2016.
- [2] Srini Janarthanam. *Hands-On Chatbots and Conversational UI Development: Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills*. Packt Publishing, 2017.
- [3] R. Pieraccini. *The Voice in the Machine: Building Computers that Understand Speech*. The MIT Press, 2012.
- [4] TechNavio. *Conversational Systems Market by Type and Geography - Forecast and Analysis 2020-2024*. TechNavio Market Research Reports, 2019.
- [5] Cathy Pearl. *Designing Voice User Interfaces: Principles of Conversational Experiences*. O'Reilly Media, 2017.
- [6] Rodrigo Bavaresco, Diórgenes Silveira, Eduardo Reis, Jorge Barbosa, Rodrigo Righi, Cristiano Costa, Rodolfo Antunes, Marcio Gomes, Clauter Gatti, Mariangela Vanzin, Saint Clair Junior, Elton Silva, and Carlos Moreira. Conversational agents in business: A systematic literature review and future research directions. *Computer Science Review*, 36:100239, 2020.
- [7] J. Liu, , L. Yanling, and L. Min. Review of intent detection methods in the human-machine dialogue system. *Journal of Physics: Conference Series*, 1267(1):012059, 2019.
- [8] Pierre Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language*, 34(1):232–255, 2015.
- [9] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review*, 21(2):97–126, 2006.
- [10] Erinc Merdivan, Deepika Singh, Sten Hanke, and Andreas Holzinger. Dialogue systems for intelligent human computer interactions. *Electronic Notes in Theoretical Computer Science*, 3434(1):57–71, 2019.
- [11] D. Suendermann and R. Pieraccini. *Crowdsourcing for Industrial Spoken Dialog Systems*. Wiley, 2013.
- [12] E. Barnard, A. Halberstadt, C. Kotelly, and M. Phillips. A Consistent Approach To Designing Spoken-dialog Systems. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'99)*, pages 1173–1176, Keystone, Colorado, USA, 1999.

- [13] R. Pieraccini and J. Huerta. Where do we go from here? Research and commercial spoken dialog systems. In *Proc. of 6th SIGdial Workshop on Discourse and Dialog*, pages 1–10, Lisbon, Portugal, 2005.
- [14] J.D. Williams. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Proc. International Conference on Spoken Language Processing (InterSpeech’08)*, pages 1173–1176, Brisbane, Australia, 2009.
- [15] M. Cohen, J. Giangola, and J. Balough. *Voice User Interface Design*. Addison Wesley, 2004.
- [16] Long Duong, Vu Cong Duy Hoang, Tuyen Quang Pham, Yu-Heng Hong, Vladislavs Dovgalecs, Guy Bashkansky, Jason Black, Andrew Bleeker, Serge Le Huitouze, and Mark Johnson. An adaptable task-oriented dialog system for stand-alone embedded devices. In *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Florence, Italy, 2019.
- [17] Rajkumar Roy, Ashutosh Tiwari, Satya Shah, and John Hadden. Soft computing in the service industry. In *Proc. of 6th International Conference on Recent Advances in Soft Computing*, pages 1–9, Canterbury, UK, 2006.
- [18] D. Griol, Zoraida Callejas, R. López-Cózar, and G. Riccardi. A domain-independent statistical methodology for dialog management in spoken dialog systems. *Computer Speech & Language*, 28(3):743–768, 2014.
- [19] J. Williams and S. Young. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):393–422, 2007.
- [20] C.J. Lee, S.K. Jung, K.D. Kim, D.H. Lee DH, and G.G. Lee. Recent approaches to dialog management for spoken dialog systems. *Journal of Computing Science and Engineering*, 4(1):1–22, 2010.
- [21] M. F. McTear. *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer, 2004.
- [22] 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(2):128–139, 2011.
- [23] S.W. Chu, I. O’Neill, P. Hanna, and M. McTear. An approach to multistrategy dialogue management. In *Proc. 9th International Conference on Spoken Language Processing (Interspeech’05-Eurospeech)*, pages 865–868, Lisbon, Portugal, 2005.
- [24] Carla Gordon, Anton Leuski, Grace Benn, Eric Klassen, Edward Fast, Matt Liewer, Arno Hartholt, and David R. Traum. PRIMER: an emotionally aware virtual agent. In *Proc. of the ACM IUI Workshops co-located with the 24th ACM Conference on Intelligent User Interfaces (ACM IUI)*, Los Angeles, USA, 2019.
- [25] S. Young, M. Gasic, B. Thomson, and J. Williams. POMDP-based Statistical Spoken Dialogue Systems: a Review. *Proc IEEE*, 101(5):1160–1179, 2013.
- [26] L.F. Hurtado, J. Planells, E. Segarra, E. Sanchis, and D. Griol. A Stochastic Finite-State Transducer Approach to Spoken Dialog Management. In *Proc. 11th Annual Conference of the International Speech Communication Association (InterSpeech’10)*, pages 3002–3005, Makuhari, Chiba, Japan, 2010.
- [27] D. Griol, L.F. Hurtado, E. Segarra, and E. Sanchis. A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication*, 50(8–9):666–682, 2008. Elsevier.
- [28] Pei-Hao Su, Milica Gasic, and Steve Young. Reward estimation for dialogue policy optimisation. *Computer Speech & Language*, 51:24–43, 2018.
- [29] S. Young, J. Schatzmann, K. Weilhammer, and H. Ye. The Hidden Information State Approach to Dialogue Management. In *Proc. 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’07)*, pages 149–152, Honolulu, Hawaii, USA, 2007.
- [30] H. Cuayáhuatl, S. Renals, O. Lemon, and H. Shimodaira. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU’05)*, pages 290–295, San Juan, Puerto Rico, 2005.
- [31] H.H. Meng, C. Wai, and R. Pieraccini. The Use of Belief Networks for Mixed-Initiative Dialog Modeling. *IEEE Transactions on Speech and Audio Processing*, 11(6):757–773, 2003. IEEE.
- [32] Jianfeng Gao, Michel Galley, and Lihong Li. *Neural Approaches to Conversational AI*. Now Publishers, 2019.
- [33] Heriberto Cuayáhuatl, Simon Keizer, and Oliver Lemon. Strategic Dialogue Management via Deep Reinforcement Learning. In *Proc. of NIPS’15 Workshop on Deep Reinforcement Learning*, pages 1–10, Montreal, Canada, 2015.
- [34] Dilek Hakkani-Tur, Gokhan Tur, Larry Heck, Ashley Fidler, and Asli Celikyilmaz. A discriminative classification-based approach to information state updates for a multi-domain dialog system. In *Proc. of 13th Annual Conference of the International Speech Communication Association*, pages 330–333, Portland, Oregon (USA), 2012.
- [35] S. Volkova, P. Choudhury, C. Quirk, B. Dolan, and L. Zettlemoyer. Lightly Supervised Learning of Procedural Dialog Systems. In *Proc. of 51st Annual Meeting of the Association for Computational Linguistics*, pages 1669–1679, Sofia (Bulgaria), 2013.

- [36] P. Cordero, M. Enciso, D. Lopez, and A. Mora. A conversational recommender system for diagnosis using fuzzy rules. *Expert Systems with Applications*, 154:113449, 2020.
- [37] Silvia Rossi and Dongheui Lee. Special issue on user profiling and behavior adaptation for human-robot interaction. *Pattern Recogn. Lett.*, 99:1–2, November 2017.
- [38] Xu Han and Tom Yeh. Evaluating voice applications by user-aware design guidelines using an automatic voice crawler. In *Proc. of the ACM IUI 2019 Workshops co-located with the 24th ACM Conference on Intelligent User Interfaces (ACM IUI)*, pages 1–4, Los Angeles, USA, 2019.
- [39] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23, 2000.
- [40] K. Scheffler and S. Young. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proc. of Human Language Technology Conference (HLT'02)*, pages 12–18, San Diego, USA, 2001.
- [41] K. Georgila, J. Henderson, and O. Lemon. Learning user simulations for information state update dialogue systems. In *Proc. of 9th European Conference on Speech Communication and Technology (Eurospeech'05)*, pages 893–896, Lisbon, Portugal, 2005.
- [42] S. Jung, C. Lee, K. Kim, D. Lee, and G.G. Lee. Hybrid user intention modeling to diversify dialog simulations. *Computer Speech and Language*, 25(2):307–326, 2011.
- [43] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. of Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 149–152, Rochester, NY, USA, 2007.
- [44] Andrew McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification. In *Proc. of AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, Madison, Wisconsin, USA, 1998.
- [45] P. Haffner, G. Tur, and J.H. Wright. Optimizing SVMs for complex call classification. In *Proc. of IEEE International Conference on Acoustics*, pages 632–635, Hong Kong, China, 2003.
- [46] Hanxiao Zhang, Wei Song, Lizhen Liu, Chao Du, and Xinlei Zhao. Query Classification Using Convolutional Neural Networks. In *Proc. of 10th International Symposium on Computational Intelligence and Design (ISCID)*, pages 632–635, Hangzhou, China, 2017.
- [47] Bing Liu and Ian Lane. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, pages 414–419, Okinawa, Japan, 2016.
- [48] Mauajama Firdaus, Ankit Kumar, Asif Ekbal, and Pushpak Bhattacharyya. A multi-task hierarchical approach for intent detection and slot filling. *Knowledge-Based Systems*, 183:104846, 2019.
- [49] K. Sreelakshmi, P.C. Rafeeque, S. Sreetha, and S. Gayathri. Deep Bi-Directional LSTM Network for Query Intent Detection. In *Proc. of IEEE 8th International Advance Computing Conference (IACC'18)*, pages 939–946, Greater Noida, India, 2018.
- [50] Asif Ekbal Pushpak Bhattacharyya Mauajama Firdaus, Hitesh Golchha. A deep multi-task model for dialogue act classification, intent detection and slot filling. *Cognitive Computation*, In Press, 2020.
- [51] Iryna Haponchyk, Antonio Uva, Seunghak Yu, Olga Uryupina, and Alessandro Moschitti. Supervised Clustering of Questions into Intents for Dialog System Applications. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2310–2321, Brussels, Belgium, 2018.
- [52] Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proc. of Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248, Lisbon, Portugal, 2015.
- [53] Dhiraj Madan and Sachindra Joshi. Finding Dominant User Utterances And System Responses in Conversations. In *Proc. of the 8th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 723–732, Taipei, Taiwan, 2017.
- [54] L. Han, L. Lushan, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. UMBC\_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proc. of Second Joint Conference on Lexical and Computational Semantics (SEM)*, pages 44–52, Atlanta, Georgia, USA, 2013.
- [55] W.H. Gomaa and A.A. Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.
- [56] Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128, 2015.

- [57] Hien T. Nguyen, Phuc H. Duong, and Erik Cambria. Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, 182:104842, 2019.
- [58] Chao Huang, Hongliang Li, Wei Li, Qingbo Wu, and Linfeng Xu. Store classification using text-exemplar-similarity and hypotheses-weighted-cnn. *Journal of Visual Communication and Image Representation*, 44:21–28, 2017.
- [59] Alberto Fernández, Paulo Peixoto, Isaac Martín, Cristina Conde, and Enrique Cabello. Combining dynamic finite state machines and text-based similarities to represent human behavior. *Engineering Applications of Artificial Intelligence*, 85:504–516, 2019.
- [60] S. Chandramohan, M. Geist, F. Lefevre, and O. Pietquin. Clustering behaviors of spoken dialogue systems users. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4981–4984, Kyoto, Japan, 2012.
- [61] P. Deepak. Mixkmeans: Clustering question-answer archives. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP’16)*, pages 1576–1585, Austin, Texas, USA, 2016.
- [62] Jean-Leon Bouraoui and Vincent Lemaire. Cluster-Based Graphs for Conceiving Dialog Systems. In *Proc. of Workshop DMNLP at European Conference on Machine Learning (ECML)*, pages 17–32, Skopje, Macedonia, 2017.
- [63] S. Bangalore, G. DiFabrizio, and A. Stent. Learning the Structure of Task-driven Human-human Dialogs. In *Proc. of 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 201–208, Sydney, Australia, 2008.
- [64] Ananlada Chotimongkol and Alexander I. Rudnicky. Acquiring Domain-Specific Dialog Information from Task-Oriented Human-Human Interaction through an Unsupervised Learning. In *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP’08)*, pages 955–964, Waikiki, Honolulu, Hawaii, 2008.
- [65] K. Zhai and J. Williams. Discovering Latent Structure in Task-Oriented Dialogues. In *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics*, pages 36–46, Lille, France, 2015.
- [66] O. Vinyals and V. Le Quoc. A neural conversational model. In *Proc. of 32nd International Conference on Machine Learning (ICML’15)*, pages 231–238, Lille, France, 2015.
- [67] Wendong Ge and Bo Xu. Dialogue Management based on Sentence Clustering. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 800–805, Beijing, China, 2015.
- [68] Romain Laroche. Content finder AssistanT. In *Proc. of 18th International Conference on Intelligence in Next Generation Networks (ICIN)*, pages 231–238, Paris, France, 2015.
- [69] Zoraida Callejas, David Griol, Klaus-Peter Engelbrecht, and Ramón López-Cózar. A clustering approach to assess real user profiles in spoken dialogue systems. In Joseph Mariani, Sophie Rosset, Martine Garnier-Rizet, and Laurence Devillers, editors, *Natural Interaction with Robots, Knowbots and Smartphones*, pages 327–334, New York, 2014. Springer.
- [70] E. Schubert, J. Sander, M. Ester, H.P. Kriegel, and X. Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *International Journal of Computer Applications*, 68(13):1–21, 2017.
- [71] David Griol and Zoraida Callejas. Discovering Dialog Rules by Means of an Evolutionary Approach. In *Proc. of 20th Annual Conference of the International Speech Communication Association (Interspeech)*, pages 1473–1477, Graz, Austria, 2019.
- [72] F.J. Ordóñez, J.A. Iglesias, P. de Toledo, A. Ledezma, and A. Sanchis. Online activity recognition using evolving classifiers. *Expert Systems with Applications*, 40(4):1248–1255, 2013.
- [73] A.W. Black, S. Burger, B. Langner, G. Parent, and M. Eskenazi. Spoken dialog challenge 2010. In *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, pages 448–453, Berkeley, CA, USA, 2010.
- [74] Lluís-F. Hurtado, Joaquin Planells, Encarna Segarra, and Emilio Sanchis. Spoken dialog systems based on online generated stochastic finite-state transducers. *Speech Communication*, 83:81–93, 2016.
- [75] A. Raux, B. Langner, A. Black, and M. Eskenazi. Doing research on a deployed spoken dialogue system: One year of Lets Go! experience. In *Proc. of International Conference on Spoken Language Processing (Interspeech’06)*, pages 65–68, Pittsburgh, Pennsylvania, USA, 2006.
- [76] J. Lopes, M. Eskenazi, and I. Trancoso. From rule-based to data-driven lexical entrainment models in spoken dialog systems. *Computer Speech and Language*, 31:87–112, 2015.

- [77] H. Hastie, N. Merigaud, X. Liu, and O. Lemon. “Let’s Go, DUDE!” Using the Spoken Dialogue Challenge to teach Spoken Dialogue development. In *Proc. of IEEE Spoken Language Technology Workshop (SLT’10)*, pages 466–471, Berkeley, USA, 2010.
- [78] 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. In *Proc. of IEEE Spoken Language Technology Workshop (SLT’10)*, pages 460–465, Berkeley, USA, 2010.
- [79] Francisco Torres, Emilio Sanchis, and Encarna Segarra. User simulation in a stochastic dialog system. *Computer Speech & Language*, 22(3):230 – 255, 2008.
- [80] Carlos-D. Martínez-Hinarejos, José-Miguel Benedí, and Ramón Granell. Statistical framework for a spanish spoken dialogue corpus. *Speech Communication*, 50(11):992–1008, 2008.