Griol, D., Molina, J.M. (2018). Building multi-domain conversational systems from single domain resources. *Neurocomputing*, 271, pp. 59-69.

DOI: 10.1016/j.neucom.2017.01.112

Elsevier Editorial System(tm) for Neurocomputing

Manuscript Draft

Manuscript Number:

Title: Building multi-domain conversational systems from single domain resources

Article Type: SI: SOCO 2015

Keywords: Dialog Systems; Multi-Domain; Spoken Interaction; Human-machine interaction; Neural Networks; Statistical Methodologies

Corresponding Author: Dr. David Griol, Ph.D.

Corresponding Author's Institution: Carlos III University of Madrid

First Author: David Griol, Ph.D.

Order of Authors: David Griol, Ph.D.; Jose Manuel Molina

Abstract: Current advances in the development of mobile and smart devices have generated a growing demand for natural human-machine interaction and favored the intelligent assistant metaphor, in which a single interface gives access to a wide range of functionalities and services. Conversational systems constitute and important enabling technology in this paradigm. However, they are usually defined to interact in semantic-restricted domains in which users are offered a limited number of options and functionalities. The design of multi-domain systems implies that a single conversational system is able to assist the user in a variety of tasks. In this paper we propose an architecture for the development of multi-domain conversational systems that allows: (1) integrating available multi and single domain speech recognition and understanding modules, (2) combining available system in the different domains implied so that it is not necessary to generate new expensive resources for the multidomain system, (3) achieving better domain recognition rates to select the appropriate interaction management strategies. We have evaluated our proposal combining three systems in different domains to show that the proposed architecture can satisfactory deal with multimodain dialogs.

Building multi-domain conversational systems from single domain resources

David Griol, José Manuel Molina

Applied Artificial Intelligence Group
Computer Science Department
Carlos III University of Madrid, Spain.
{*david.griol,josemanuel.molina*}*@uc3m.es*

Abstract

Current advances in the development of mobile and smart devices have generated a growing demand for natural human-machine interaction and favored the intelligent assistant metaphor, in which a single interface gives access to a wide range of functionalities and services. Conversational systems constitute and important enabling technology in this paradigm. However, they are usually defined to interact in semantic-restricted domains in which users are offered a limited number of options and functionalities. The design of multi-domain systems implies that a single conversational system is able to assist the user in a variety of tasks. In this paper we propose an architecture for the development of multi-domain conversational systems that allows: (1) integrating available multi and single domain speech recognition and understanding modules, (2) combining available system in the different domains implied so that it is not necessary to generate new expensive resources for the multi-domain system, (3) achieving better domain recognition rates to select the appropriate interaction management strategies. We have evaluated our proposal combining three systems in different domains to show that the proposed architecture can satisfactory deal with multi-domain dialogs.

Keywords: Dialog Systems, Multi-Domain, Spoken Interaction, Human-machine interaction, Neural Networks, Statistical Methodologies

1. Introduction

A spoken dialog system (SDS) is a software that accepts natural language as an input and produces natural language as an output engaging in a conversation with the user [1, 2, 3]. To successfully manage the interaction with users, spoken dialog systems usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS). These tasks are usually implemented in different modules.

Recent advances in conversational interfaces has been propelled by the convergence of three enabling technologies. First, the Web emerged as a universal communications channel. Web-based dialog systems are scalable enterprise systems that leverage the Internet to simultaneously deliver dialog services to large populations of users. Second, the development of mobile technologies and intelligent devices, such as smartphones and tablets, have made it possible to deploy a large number of sensors and to integrate them into dialog systems that provide multimodal interaction capabilities (i.e., use of different modalities for the input and/or output of the system) and allow their access in almost every place and at any time. Third, computational linguistics, the field of artificial intelligence that focuses on natural language software, has significantly increased speech recognition, natural language understanding and speech synthesis capabilities [1, 3].

These advances have extended the initial application domains of dialog systems to complex information retrieval and question answering applications [4], surveys applications [5], e-commerce systems [6], recommendations systems [7], e-learning and tutoring systems [8], in-car systems [9], spoken dialog within vehicles [10], remote control of devices and robots in smart environments [11], Ambient Assisted Living systems [12], or virtual companions [13].

However, spoken dialog systems are usually employed within single domains, which are also defined according a static set of strong restrictions. In mobile environments, the dynamic support for a wide range of topics and multiple tasks within one and the same dialog is still a major challenge as people require increasingly more functionalities in the system [14, 15].

As mobile and smart devices and wearables become widespread and the number of functionalities and services they provide grows, there appears an increasingly urgent need for ways of tackling this diversity in a way that is transparent to the user. This explains why the main providers of mobile operating systems are working towards developing assistants (e.g. Apple Siri, Google Now or Microsoft Cortana) to meet this growing demand. Although such systems are endowed with the capacity to understand and synthesize speech, their conversational capabilities are very restricted and are usually employed for question and answer exchanges that involve isolated user-system turns. In order to be able to show a complex conversational behavior in different domains, a more complex approach is required.

In this paper we contribute a novel approach to generate multi-domain conversational systems that are able to hold a conversation in which the user switches from a domain to the other. Section 2 presents a state of the art of the most relevant approaches that exist to address this challenge and how they relate with our proposal. Section 4 describes our proposal in detail. Section 4 presents the experimental set-up to show the applicability of our proposal. Section 5 discusses the results of the evaluation for the developed multi-domain dialog system, and finally Section 6 presents conclusions and outlines possibilities for future work.

2. State of the art

The most widespread approach to develop multi-domain dialog systems is based on the use of distributed architectures [16, 17, 18]. These architectures are based on using a single interface with specific dialog managers to deal with each domain or set of related tasks. Selecting the most appropriate domain is then one of the key technologies to develop a multi-domain dialog system [15]. Gasic et al. have recently proposed a distributed multi-domain dialog architecture in which dialog policies are organized in a class hierarchy aligned to an underlying knowledge graph [19]. Gaussian process-based reinforcement learning is proposed to construct generic dialog policies. A policy committee model, based on a Bayesian committee machine (BCM) is proposed in [20] to further improve the performance when the data is limited.

To solve the domain identification problem in a multi-domain dialog system, the initial approach was related to asking users to explicitly specify the domain [21]. Although this approach prevents the problem of ambiguity caused from polysemous words, the dialog strategy is not natural and users have to know each domain in advance [22]. The distributed architecture described in [18] proposes domain identification computing the most similarity scores of the recognized user utterance and the grammar/language model of domains. Although it is an implicit domain identification approach, the identification of ambiguous domains is still not solved because the similarity scores could be almost the same for polysemous words.

Pre-selection or post-selection methodologies have been proposed for domain selection. Preselection methodologies select the most appropriate dialog manager by considering the features extracted from the user utterance. In this approach, a specific module is included to parse the user turns and redirect them to the appropriate single-domain dialog manager [16, 18, 23]. This approach is efficient in execution time, but requires incorporating domain-specific knowledge to improve the domain selection process [24, 15].

The domain selection process proposed in [14] is based on a Logistic Regression (LR) model to classify the current user turn into a specific domain. The features used for the vectorization of the sentence are: bag of words, bag of bigrams, and co-occurrence of two words in the same sentence. A one-vs-all classifier was created for each subtask that assigned a score to the input sentence. The task with highest confidence is usually selected; however, if the confidence of the second-ranked task is close, the Task Manager outputs a disambiguation turn asking the user to add some domain-related expression.

A series of domain-independent analyzers including linguistic analysis, generic spoken language understanding (SLU) analysis, and keyword analysis is proposed in [15]. Based on the analyzed results, domain selection is performed by two-step approaches: domain ordering and domain filtering. In domain-ordering step, the domain candidates are listed in descending order of scores computed by a preselection model. Then, content-based domain filtering is performed for each domain in order to determine the selected domain.

A two-level SLU approach is proposed in [25], where, at first level, the result of ASR is tagged and classified by general SLU for domain spotting by employing a maximum entropy-based classifier using lexical word, dialog acts and previous domain as classification features. The use of two ontologies is proposed in [26]: one associated with a broad-coverage SLU module, and a second associated with a task-domain. The use of Recurrent Neural Networks has been very recently proposed in [27, 28].

Post-selection methodologies are based on the results provided by the different dialog managers integrated in the multi-domain system [29]. The main advantage of this approach is that rich domain-specific features are considered to improve performance. However, executing all the dialog managers on the unrelated domains can be a waste of time especially when the number of domains increases [15].

Some techniques for multi-domain semantic speech recognition and understanding and language generation have been presented during recent years [30, 31, 32, 33, 34]. In the case of dialog management, the exponential increase of dialog states makes it really difficult to create a dialog manager that can serve several domains using the approaches that are used for single-domain systems [14].

As described in [22], there are two major difficulties for multi-domain dialog management. The first one is to interpret users' interested domain correctly given ambiguous user utterances across different domains. The second one is the high cost of merging the dialog management of different single-domain systems into one multi-domain system.

Frame-based representations are usually employed to model the semantic representation of the user' contents in recent proposals to develop multi-domain dialog systems [35, 29, 24]. The form interpretation algorithm (FIA), the basis for the VoiceXML standard[1], is an example of a model of frame-based dialog manage-

[1] https://www.w3.org/TR/voicexml20/

ment. The use of this standard is proposed in [36] to implement multi-domain dialog systems in which a content manager automatically extracts the contents for each domain from the Internet and a content spotter selects the most appropriate source by means of the cosine similarity function.

The “Information State” theory [37] represents a dialog by means of the information required to differentiate it from other dialogs. This information, which is also referred as the “discourse context” or the “mental state” represents the effects of the sum of previous actions during the dialog and the motivation for the future system actions. According to this theory, the main tasks of an Information State dialog manager are to update the information state based on the observed user actions, and then select the next system action. This approach was used to develop the GodiS architecture, which is proposed in [38] to develop multi-domain dialog systems. This approach is adopted in [39] to develop a multi-domain SDS that uses a discriminative classification model for more accurate state updates.

Agent-based dialog management approaches combine the benefits of finite-state and frame-based dialog management approaches [40]. The dialog managers developed by means of this approach allow to execute and monitor operations in dynamically changing application domains. Similarly, the developed dialog managers can benefit from the mixed-initiative approach [41]. The RIME framework (Robot Intelligence based on Multiple Experts) employs this approach to integrate different dialog agents, which are specialized in achieving specific tasks by means or performing physical actions or engaging the user in a dialog [42].

Different example-based dialog management approaches have been recently proposed to develop multi-domain dialog systems [43, 44, 24, 45]. The dialog managers developed by means of these approaches employs a database of pairs of a dialog example and the corresponding system action. As described in different contributions [46, 47, 24], the dialog managers developed by means of this approach can be easily and flexibly modified by updating dialog examples in the database, which is specially important to construct multi-domain dialog managers when the specific domains or tasks can be frequently expanded or when the limited knowledge about the task makes very difficult to define rule-based managers.

However, a large number of dialog examples is required to cover the variety of inputs in the dialog and avoid the situations in which no dialog examples are available in the database for the specific dialog state. To solve this problem, the use of dialog corpora automatically acquired from the Web (e.g. social networks or movie databases) as dialog examples has been proposed in recent proposals [46, 47]. The use of an active learning framework to reduce the effort required to construct example bases for each specific domain of a dialog system has been recently proposed in [43].

During recent years, statistical approaches for dialog management have been proposed to automate the processes for developing, deploying and re-deploying conversational applications and also reduce the time-consuming process required to manually define rule-based dialog managers [48, 49, 50, 51]. In addition, statistical dialog managers allow to maintain a probabilistic distribution over many hypotheses corresponding to system actions for the current dialog state.

The main trend in this area is an increased use of dialog corpus to learn optimal dialog strategies using machine-learning methods to develop dialog systems with more robust performance, improved portability, better scalability, and easier adaptation to other tasks. The use of training corpus to develop statistical dialog managers allow to explicitly model the variance in user behaviors and preferences that can be difficult to address by means of hand-written rules [52]. However, as described in [53], most of the existing multi-domain spoken dialog systems in public use rule-based dialog management methodologies [17], and the application of statistical models in multi-domain dialog management approaches is still preliminary.

3. Our proposal for developing multi-domain dialog systems

Figure 1 shows the complete architecture of a dialog system integrating our proposal for multi-domain dialog support. As can be observed, the architecture consists of a set components, linked together by communication channels, that can be classified into Input and Output Components, Domain Detection Components, and Natural Language Processing and Dialog Components.

User inputs are processed in the proposed architecture by the *Text Inputs* and the *Automatic Speech Recognition* (ASR) Components. The *Text Inputs* Component allows to collect text inputs provided by means of the keyboard and tactile screen of portable devices. Moreover, it can subscribe to any channel and displays the received message as plain text.

The goal of speech recognition is to obtain the sequence of words uttered by a speaker. It is a very complex task, as there can be a great deal of variation in the input the recognizer must analyze, for example, in terms of the linguistics of the utterance, inter and intra speaker variation, the interaction context and the transmission channel [54]. In our proposal, the vocabulary of the ASR has to cover all the vocabularies of the different sub-domains that are included. There are a number of different open-source tools as well as commercially available products that allow developers to add domain-independent speech input and output to their applications. For the Web, we recommend HTML5 Web Speech API (Web SAPI)[2],

[2] https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-tothe-Web-Speech-API?hl=en

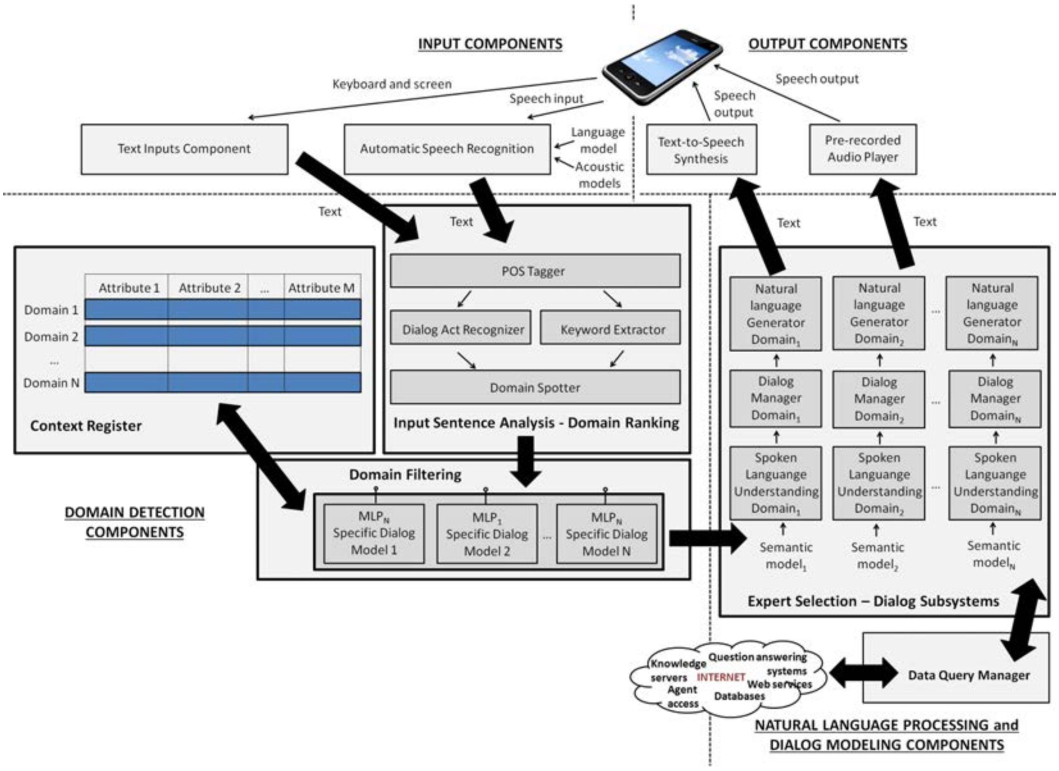
INPUT COMPONENTS
OUTPUT COMPONENTS
Keyboard and screen
Speech input
Speech output
Speech output
Text Inputs Component
Automatic Speech Recognition
Language model
Acoustic models
Text-to-Speech Synthesis
Pre-recorded Audio Player
Text
Text
Text
Text
Attribute 1
Attribute 2
...
Attribute M
Domain 1
Domain 2
...
Domain N
Context Register
POS Tagger
Dialog Act Recognizer
Keyword Extractor
Domain Spotter
Input Sentence Analysis - Domain Ranking
Domain Filtering
MLP$_N$ Specific Dialog Model 1
MLP$_1$ Specific Dialog Model 2
...
MLP$_N$ Specific Dialog Model N
DOMAIN DETECTION COMPONENTS
Natural language Generator Domain$_1$
Natural language Generator Domain$_2$
...
Natural language Generator Domain$_N$
Dialog Manager Domain$_1$
Dialog Manager Domain$_2$
...
Dialog Manager Domain$_N$
Spoken Languange Understanding Domain$_1$
Spoken Languange Understanding Domain$_2$
...
Spoken Languange Understanding Domain$_N$
Semantic model$_1$
Semantic model$_2$
Semantic model$_N$
Expert Selection – Dialog Subsystems
Knowledge servers
Question answering systems
INTERNET
Web services
Agent access
Databases
Data Query Manager
NATURAL LANGUAGE PROCESSING and DIALOG MODELING COMPONENTS

Figure 1: Proposed modular architecture to develop multi-domain dialog systems

while for developing conversational mobile devices we recommend Android Speech APIs [3]. Other recommended open-source tools for speech recognition are AT&T speech API [4], CMU Sphinx[5], and HTK Cambridge[6].

Once the dialog system has recognized what the user uttered, it is necessary to understand what they said. Spoken language processing generally involves morphological, lexical, syntactical, semantic, and pragmatical knowledge [55]. The dialog manager decides then the next action of the system [50], interpreting the incoming semantic representation of the user input in the context of the dialog. In addition, it resolves ellipsis and anaphora, evaluates the relevance and completeness of user requests, identifies and recovers from recognition and understanding

[3] http://developer.android.com/reference/android/speech/SpeechRecognizer.html

[4] http://developer.att.com/apis/speech

[5] http://cmusphinx.sourceforge.net/

[6] http://htk.eng.cam.ac.uk/ (Accessed May 5, 2016).

errors, retrieves information from data repositories, and decides about the next system's response. Natural language generation is the process of obtaining sentences in natural language from the non-linguistic, internal representation of information handled by the dialog system [56]. Finally, the TTS module transforms the generated sentences into synthesized speech [57]. One of the main advantages of our proposal to develop a multi-domain dialog system is that it integrates the Spoken Language Understanding, Dialog Manager, Natural Language Generator, and TTS modules of each one the domain-dependent subsystems.

In our proposal the domain of the current user utterance is identified in two steps. Firstly, a domain ranking is established combining keyword-based and features-based approaches, the resulting ordered list of domains is filtered in a second stage to decide the most appropriate one. The second stage uses a data structure that stores the complete history of the dialog and a statistical model generated for each domain that is based on it. The data structure is updated by each of the single-domain systems once they are selected for a particular user turn as is inputed sequentially to the statistical model corresponding to each domain in the ordered list until one of them produces an acceptable output. When this happens, the domain is selected and the corresponding single-domain system is used to manage the current user turn. The following subsections describe this process in more detail.

3.1. Domain Ordering: The input sentence analysis

The methodology that we propose to develop the domain detector module is based on the proposals described in [24, 14]. Our proposal combines keyword-based and feature-based approaches for automatically classifying the domains. The keyword spotting technique and the feature-based classification techniques are used independently to develop a spotter module as described in [24].

The features used for the domain spotter include: Linguistic features (bag of words, POS tags, bag of bigrams, and co-occurrence of two words in the same sentence), semantic features (dialog acts), and Keyword features extracted using the traditional keyword spotting method (n-Best keyword and n-best domain class).

To extract the keyword features, we extract more obvious keywords which are assigned to the weight correlated to the particular domain class using the term weighting methods [24]. For automatic extraction of keywords from the dialog corpora, we apply term frequency and inverse document frequency (TF * IDF) [58] and salience measure [59] as term weighting methods. Although the linguistic features impose the keyword information using the bag-of-words, the keyword features are more indicative and reliable features than the pure bag-of-words to improve the performance of the domain identification.

As proposed in [14], the domain spotter uses a Logistic Regression (LR) model [60] to classify the current user turn into a specific domain. A “one-vs-all” classifier was learned for each specific domain to assign a score to the input sentence. In the “one-vs-all” or OVA [61] scheme, N different binary classifiers are used, each one trained to distinguish the examples in a single class from the examples in all remaining classes. The N classifiers are run to classify each input x, and the classifier which outputs the largest (most positive) value is chosen. For the $i-th$ classifier, let the positive examples be all the points in class i, and let the negative examples be all the points not in class i. Let f_i be the $i-th$ classifier. The classification function is as follows

$$f(x) = \operatorname*{argmax}_i f_i(x)$$

To improve the operation of this classification technique, instead of always selecting the domain with highest confidence, if the confidence of the second-ranked domain is close, the Expert Selection informs informs the dialog manager to select a disambiguation turn asking for the user’s goal. The main objective of these system turns is to make users add some task-related expressions like “train” or ”weather” that can be used by the task classifier to make more accurate predictions.

3.2. Domain Filtering: Improving the domain identification

We propose a second step to improve domain identification. A *Context Register* (*CR*) is defined to store the information about the multi-domain dialogs considering the list of dialog acts defined for each subsystem. The Spoken Language Understanding module of each system is used to extract the information that is relevant to the corresponding domain and incorporate this information to the *Context Register*.

To estimate the probability of the different systems to continue the dialog we have adapted a recently developed dialog management methodology [50] to deal with multi-domain dialogs. According to this methodology, a dialog is represented 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

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

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

At time i, the objective of the dialog manager of each subsystem is to find the best system answer A_i that allows to continue the dialog. This selection is a local process for each time i and takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog preceding time i:

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

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

The main problem to resolve this equation is regarding the number of possible sequences of states, which is usually very large. To solve the problem and extend the proposed methodology for multi-domain dialog management, we propose the use of the *Context Register* as a data structure to store the information provided by the user throughout the previous history of the dialog. As previously describes, the contents of the CR are updated by the SLU module of the specific subsystem that is selected to generate the next system response. The selection of the best system response A_i is then given by:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | CR_{i-1}, S_{i-1}) \tag{2}$$

As propose to solve Equation 2 by means of a classification process. This way, every dialog situation (i.e., each possible sequence of dialog acts) is classified taking into account a set of classes $\mathcal{C}$, in which a class contains all the sequences that provide the same set of system actions (responses). The objective of the dialog manager at each moment is to select a class of this set $c \in \mathcal{C}$, so that the system answer is the one associated with the selected class.

The classification function can be defined in several ways. In the previous evaluation of single-domain dialog systems [50], the best results were obtained using a multilayer perceptron (MLP) [62] where the input layer holds the input pair (CR_{i-1}, S_{i-1}) corresponding to the Context Register and the state. The values of the output layer can be seen as an approximation of the a posteriori probability of the input belonging to the associated class $c \in \mathcal{C}$.

For the multi-domain dialog management, the CR is defined as a matrix structure in which the number of rows is equal to the number of subsystems that are combined and the number of columns is equal to the sum of concepts of attributes defined for each one of them. At the beginning of the dialog, all the positions in the CR are empty. During the dialog, the positions in the CR are updated to store the value provided for the SLU module of the corresponding system and the number of system turn in which it has been included.

4. Experimental set-up

To evaluate our proposal, we have used our approach to build a multi-domain dialog system from three already existing systems in different domains: travel planning, weather forecast and touristic information. We have chosen the domains so that they are not exclusive, as identifying the appropriate domain to respond to each user turn would have been very straightforward in disparate domains. On the contrary, the domains selected share many concepts (e.g. cities are relevant for the three domains) and thus the task of selecting the best domains for each input is not obvious.

4.1. The TRAVEL-PLANNING task

The dialog system developed for the DIHANA project provides railway information [63]. The *Interaction Register* defined for the task consists of the possible queries that users can make (*Timetables-Query*, *Fares-Query*, *Train-Type-Query*, *Trip-Time-Query*, or *Services-Query*) and ten attributes that they must provide to complete these queries (*Origin*, *Destination*, *Departure-Date*, *Arrival-Date*, *Departure-Hour*, *Arrival-Hour*, *Ticket-Class*, *Train-Type*, *Order-Number*, and *Services*). Users can also provide three task-independent dialog acts (*Affirmation*, *Negation*, and *Not-Understood*).

A total of 51 system responses were defined, which can be classified into questions to require data from the user (e.g., *Question:Departure-Date*, *Question:Departure-Hour*), confirmations of concepts and attributes (e.g., *Confirmation:Train-Type*, *Confirmation:Ticket-Class*), answers obtained after a query to the database (e.g., *Answer:Timetables*, *Answer:Fares*), and responses related to dialog formalities (e.g., *0pening*, *Closing*, *Acceptance*, *Rejection*, *Waiting*, *Not-understood*, etc.).

A set of 900 dialogs was acquired in the project by means of the Wizard of Oz technique [64], which allows the acquisition of a dialog corpus by means of of users interacting with an interface without knowing that the responses are being generated by a human, not a real spoken dialog system. These dialogs were used to train the statistical models for the SLU and dialog management modules of the system. Figure 2 shows an example of a dialog for the railway information system. Turns with S refer to system turns, and turns with U refer to user turns.

4.2. The WEATHER task

We also developed a dialog system to give users easy access to weather forecast information in Spain by means of spoken natural language. This information is automatically obtained from the website of the Spanish Meteorological Agency

S_1: Welcome to the system. How can I help you?
(Opening)
U_1: I want to know timetables from Madrid to Valencia.
(*Timetables-Query*) *Origin*:Madrid *Destination*:Valencia
S_2: Tell me the departure date.
(*Question:Departure-Date*)
U_2: *Next Tuesday during the morning.*
Departure-Date: 08/05/2016 *Departure-Hour*: 07.00-14.00
S_3: There are two trains. The first one leaves at 10.00 and the second one leaves at 11.30. Do you want anything else?
(*Answer:Timetables*) (*Question:New-Query*)
U_3: *Yes, the price for the first train in standard class.*
(*Affirmation*) *Question:Fares Order-Number*: First *Ticket-Class*: Standard
S_4: The price is 57.9 euros. Anything else?
(*Answer:Fares*) (*Question:New-Query*)
U_4: No, thank you.
(*Rejection*)
S_5: *Thanks for using this service. Have a good trip.*
(*Closing*)

Figure 2: Example of a dialog for the TRAVEL-PLANNING task

(AEMET[7]). The semantics of this task follow the definitions for the CUED dialog acts [65] and it is similar to the one defined for the weather information domain by similar systems [24, 14, 15].

The system allows users to ask for information (*request*) or to confirm if something is a specific weather forecast is probable (*confirm*). The information that is provide by the system includes general weather conditions (e.g., sunny, windy, cloudy), temperature ranges, precipitation probabilities, humidity, visibility, pressure, wind direction, wind speed, pollution (in the range Low to Very High), strength of the ultraviolet (UV) radiation (in the range 1 to 8), and pollen (in the range Low to Very High).

The *Interaction Register* for the task includes the possible queries that users can make (*Weather-State-Query*, *Temperatures-Query*, *Rains-Query*, *Humidity-Query*, *Visibility-Query*, *Pressure-Query*, *Wind-Direction-Query*, *Pollution-Query*, *Wind-Speed-Query*, *UV-Query*, *Weather-State-Confirmation*, *Temperatures-Confirmation*, *Rains-Confirmation*, *Humidity-Confirmation*, *Visibility-Confirmation*, *Pressure-Confirmation*, *Wind-Direction-Confirmation*, *Wind-Speed-Confirmation*, *UV-Confirmation*, or *Pollution-Confirmation*) and the attributes to complete these queries (*Location*, *Dates*, *Hours*, and *Weather-State*). Users can also provide three task-independent dialog acts (*Affirmation*, *Negation*, and *Not-Understood*).

The system responses can be classified into questions to require data from

[7] http://www.aemet.es

the user (e.g., *Question:Dates*, *Question:Location*), confirmations of concepts and attributes (e.g., *Confirmation:Weather-State*, *Confirmation:Dates*), answers obtained after a query to the database (e.g., *Answer:Rains-Query*, *Answer:Weather-State*), and responses related to dialog formalities (e.g., *0pening*, *Closing*, *Acceptance*, *Rejection*, *Waiting*, *Not-understood*, etc.).

A set of 300 human-human dialogs was acquired for the task to train the statistical models for the ASR and SLU modules of the system. Figure 3 shows an example of a dialog for the WEATHER task.

S_1: Welcome to the system. How can I help you?
(*Opening*)
U_1: I want to know the temperature for tomorrow morning in Valencia.
(*Temperatures-Query*) *Location*:Valencia *Dates*:11/05/2016 *Hours*:07.00-14.00
S_2: The temperature for tomorrow morning is expected to be high. Around 18 degrees at 07.00 and 26 degrees at 14.00. Anything else?
(*Answer:Temperatures*) (*Question:New-Query*)
U_2: *It is going to rain?*
Rains-Query
S_3: There is a chance of isolated downpours mainly across western areas at the end of the morning. Do you want anything else?
(*Answer:Rains-Query*) (*Question:New-Query*)
U_3: No, thank you.
(*Rejection*)
S_4: *Thanks for using this service. Have a good day.*
(*Closing*)

Figure 3: Example of a dialog for the WEATHER task

4.3. The TOURIST-INFORMATION task

The third practical system that we have integrated for the practical application of our proposal provides tourist information for different cities in Spain [66] The information provided is related to interesting spots and monuments, restaurants and bars, theater listings, and movie showtimes. This information is retrieved from different web repositories, most of them updated daily.

The system allow users to complete 5 types of queries (*Interesting-Spots*, *Hotel-Booking*, *Restaurants-Bars*, *Theater-Listings*, and *Movie-Showtimes*). The attributes required by the system to generate a response to the different user queries and the different system responses are shown in Table 1. A total of 56 system actions (DAs) were defined taking into account the information that the system provides, asks or confirms. Users can also provide 3 task-independent dialog acts (*Affirmation*, *Negation*, and *Not-Understood*).

A set of 200 dialogs was acquired with the system by means of its interaction with 20 recruited users [66]. These users followed a set of scenarios that specified a

Query	Attributes	System responses
Interesting-Spots	*City*	*Ask-City, Confirm-City, Provide-Sightseeing*
Hotel_Booking	*City*	Ask-City, Confirm-City
	Hotel_Name	Ask-Hotel_Name, Confirm-Hotel_Name
	Hotel_Category	Ask-Hotel_Category_Name, Confirm-Hotel_Category
	Check_in_Date	Ask-Check_in_Date, Confirm-Check_in_Date
	Check_out_Date	Ask-Check_out_Date, Confirm-Check_out_Date
	Number_Rooms	Ask-Number_Rooms, Confirm-Number_Rooms
	Number_People	Ask-Number_People, Confirm-Number_People
Theater_Listings *Movie_Showtimes*	*City*	Ask-City, Confirm-City
	Category	Ask-Category, Confirm-Category
	Show	Ask-Show, Confirm-Show
	Theater	Ask-Theater, Confirm-Theater
	Cinema	Ask-Cinema, Confirm-Cinema
	Date	Ask-Date, Confirm-Date
	Hour	Ask-Hour, Confirm-Hour
		Provide-Theater_Listings
Restaurants_Bars	*City*	Ask-City, Confirm-City
	Type	Ask-Type, Confirm-Type
	Price	Ask-Price, Confirm-Price
		Provide-RestaurantsBars_Information

Table 1: Semantic representation defined to model the TOURIST-INFORMATION task

set of objectives that must be fulfilled by the user at the end of the dialog. Figure 1 shows an example of a dialog extracted from this corpus.

S_1: Welcome to the system. How can I help you?
(*Opening*)
U_1: I want to know the information about the most interesting places to visit in Valencia.
S_2: I recommend you to visit the following places in Valencia ... Anything else?
(*Answer:Interesting-Places*) (*Question:New-Query*)
U_2: *Can you tell me the theater listings for El Musical theater today?*
(*Answer:Interesting-Places*) (*Question:New-Query*)
S_2: For the El Musical theater, there is a play called Cabanyal Intim today at 22.00... Anything else?
U_3: No, thank you.
(*Rejection*)
S_4: *Thanks for using this service. Have a good day.*
(*Closing*)

Figure 4: Example of a dialog for the TOURIST-INFORMATION task

5. Evaluation

This section describes the process followed for the evaluation of our proposal with the previously described experimental set-up. We have completed an evaluation of the domain ranking and filtering processes, and an overall assessment of the complete proposed architecture with simulated and real users.

5.1. Evaluation of the input sentence analysis and the domain ranking step

Before evaluating the complete proposal for the development of multi-domain dialog systems, we assessed the correct operation of the input sentences analysis and domain ranking. We trained the described model using a total of 2461 sample sentences extracted from the dialogs and the training samples of the Spoken Language Understanding modules of each domain. A test set of 492 sample sentences was used for the evaluation.

Table 2 shows the results of the evaluation of the proposed input sentence analysis for domain ranking. As it can be observed, the best results were obtained using TF * IDF. This result shows that the use of keyword features is helpful to improve the performance of domain identification and that the spotter module is able of detecting each specific domain. The errors encountered most often corresponded to turns where there was no proper task information or to turns that were valid for more than one task (for example, "not" or "next Friday").

Baseline (only linguistic features)	83.67%
+ Semantic features	86.37%
+ Keyword features (TF * IF)	89.72%
+ Keyword features (Salience)	87.91%

Table 2: Evaluation of the input sentence analysis and domain ranking

5.2. Evaluation of the domain filtering step

The same training and test partitions were used to evaluate our proposal for domain filtering. As it have been described in Section 3.2, the Context Register is considered to take the previous history of the dialog into account for the evaluation of the proposed classification-based methodology. Table 3 shows the results of the evaluation considering only the use of the proposed domain filtering technique for the domain detection, an also the combination of the domain ranking and domain filtering techniques as explained in Section 3.2. As it can be observed, the combination of both techniques allows to improve the domain detection rate in a 3.7% absolute.

Using only domain filtering	88.67%
Domain Ranking + Domain filtering	92.37%

Table 3: Evaluation of the domain filtering

5.3. Overall evaluation of our proposal

A previously developed statistical user simulation technique [67] has been used to complete a detailed evaluation of our proposal. The use of this kind of techniques has been already used in previous work to evaluate proposals to develop multi-domain dialog systems [20, 27, 68, 29, 24].

The statistical user simulation technique allows to evaluate a dialog system by means of the automatic generation of a number of dialogs with reduced effort. The semantics for the user model is defined by means of the list of user dialog acts defined for the dialog systems that are combined in the multi-domain approach. The user model takes into account the user dialog acts that have provided by the model until the current state of the dialog, the last system response, and the objective(s) specified for the dialog. These objectives are defined by means of dialog scenarios. Table 4 shows the complete set of scenarios defined for the practical evaluation of our proposal for the described domains.

Scenario	**Description**
Travel-Planning1	Obtain timetables for a specific origin, destination, and date.
Travel-Planning2	Obtain fares for a specific origin, destination, date, and type of train.
Travel-Planning3	Obtain timetables and fares for a specific origin, destination, and date.
Travel-Planning4	Obtain timetables and fares for a specific origin, destination, date, and hour.
Travel-Planning5	Obtain timetables and fares for a specific origin, destination, date, and type of train.
Weather1	Obtain weather forecast for a specific city and dates.
Weather2	Confirm weather forecast for a specific city and dates.
Weather3	Obtain weather forecast and temperatures for a specific city and dates.
Weather4	Confirm weather forecast and obtain temperatures for a specific city and dates.
Weather5	Confirm weather forecast and obtain pollution and UV levels for a specific city and dates.
Tourist-Information1	Obtain interesting spots for a specific city.
Tourist-Information2	Book an hotel for a specific city and dates (rest of attributes are optional).
Tourist-Information3	Book a ticket for a specific theater or cinema, city and dates (rest of attributes are optional).
Tourist-Information4	Obtain information for a specific type of restaurants/bars (rest of attributes are optional).
Tourist-Information5	Tourist-Information1 + Tourist-Information2.

Table 4: Set of scenarios defined for the evaluation of the dialog systems by means of the user simulation technique

Firstly, we completed a comparative evaluation of each one of the systems developed to deal with a single domain and the multi-domain system developed by means of our proposal. A total of 200 dialogs were simulated for each one of the scenarios defined for each sub-domain dialog system. Table 5 shows the percentage of successful simulated dialogs and average number of turns for each scenario. As it can be observed, the multi-domain dialog system achieves similar values that the single-domain systems for the complete set of scenarios.

Use of each specific dialog system					
Travel-Planning1	96.1% - 5.2	Weather1	99.7% - 4.3	Tourist-Information1	99.6% - 3.1
Travel-Planning2	94.3% - 5.6	Weather2	99.5% - 4.5	Tourist-Information2	92.1% - 6.9
Travel-Planning3	93.1% - 6.3	Weather3	98.1% - 5.6	Tourist-Information3	91.3% - 7.6
Travel-Planning4	93.0% - 6.7	Weather4	97.6% - 5.8	Tourist-Information4	96.2% - 4.2
Travel-Planning5	92.6% - 7.3	Weather5	88.1% - 7.9	Tourist-Information5	94.3% - 8.8
Use of the multi-domain dialog system					
Travel-Planning1	96.1% - 5.2	Weather1	99.7% - 4.3	Tourist-Information1	99.6% - 3.1
Travel-Planning2	94.3% - 5.6	Weather2	99.4% - 4.5	Tourist-Information2	92.1% - 6.9
Travel-Planning3	92.8% - 6.4	Weather3	97.9% - 5.8	Tourist-Information3	90.8% - 7.7
Travel-Planning4	92.9% - 6.7	Weather4	97.6% - 5.8	Tourist-Information4	96.2% - 4.2
Travel-Planning5	92.1% - 7.5	Weather5	86.9% - 8.3	Tourist-Information5	93.9% - 9.1

Table 5: Results of the comparative evaluation of the single dialog systems and the multi-domain system

Then, we evaluated the operation of our proposal for the combination of these scenarios to define multi-domain dialogs. Table 6 shows the results of the evaluation for a set of combined scenarios. As it can be observed, the multi-domain system obtains satisfactory results in the identification of the different domains and the achievement of the objectives defined in the combinations of scenarios.

Combination of two scenarios					
TP1+W1	98.7% - 6.4	TP1+TI1	97.3% - 5.5	W1+TI1	99.6% - 4.7
TP2+W2	97.1% - 6.8	TP2+TI2	92.6% - 7.2	W2+TI2	93.21% - 7.1
TP3+W4	93.1% - 7.1	TP3+TI3	90.8% - 8.0	W4+TI2	92.1% - 7.2
TP5+W5	87.3% - 8.2	TP5+TI5	93.6% - 9.1	W4+TI5	93.9% - 9.1
Combination of three scenarios					
TP1+W1+TI1	97.1% - 6.8	TP1+W2+TI2	96.2% - 7.1	TP3+W1+TI1	92.6% - 7.2
TP3+W2+TI2	91.0% - 7.8	TP4+W3+TI2	90.9% - 7.7	TP5+W4+TI5	89.1% - 9.6

Table 6: Results of the evaluation combining scenarios (TP:Travel-Planning, W:Weather, TI:Tourist-Information)

5.4. Evaluation with real users

Finally, we evaluated the behavior of the multi-domain system with real users using a subset of the scenarios designed for the user simulation. A total of 150 dialogs were recorded from interactions of six users employing the system. The evaluation was carried out by students and lecturers in our department following the types of scenarios described in the paper in different settings . An objective and subjective evaluation were carried out. We considered the following measures for the objective evaluation: i) Dialog success rate; ii) Average number of turns per dialog (nT); iii) Confirmation rate. It was computed as the ratio between

Success Rate	94%
Average Number of Turns	7.8
Confirmation Rate	25%
%ECR	86%
nCE	0.92
nNCE	0.16

Table 7: Results of the objective evaluation of the multi-domain dialog system with real users

the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT); iv) Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager; v) Average number of uncorrected errors per dialog (nNCE). This is the average of errors not corrected by the dialog manager. Only errors that modify the values of the attributes are considered; vi) Error correction rate (%ECR). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table 7 show that in most cases the multi-domain system has the capability of correctly interacting with the user. The values obtained for the average number of turns were slightly higher than the ones obtained with the user simulation, as in some dialogs the real users provided additional information which was not mandatory for the corresponding scenario or asked for additional information not included in the definition of the scenario once its objectives were achieved. The main problem detected was related to user utterances misrecognized with a very high ASR confidence. This erroneous information was annotated in the Context Register and forwarded to the specific dialog managers. However, as the success rate shows, this fact did not have a considerable impact on the system operation.

In addition, we asked the users to complete a questionnaire to assess their subjective opinion about the system performance. The questionnaire included the five questions described in Table 8. The possible answers for each one of the questions were the same: Never/Not at all, Seldom/In some measure, Sometimes/Acceptably, Usually/Well, and Always/Very Well. All the answers were assigned a numeric value between one and five. From the results, it can be observed that the multi-domain system is considered to correctly understand the different user queries, facility of obtaining the data required to fulfill the complete set of objectives of the scenario, and the suitability of the interaction rate during the dialog.

Question	Mean	Standard deviation
How well did the system understand you?	4.3	0.4
How well did you understand the system messages?	4.4	0.5
Was it easy for you to get the requested information?	4.6	0.4
Was the interaction rate adequate?	4.3	0.4
Was it easy for you to correct the system errors?	3.7	0.6
In general, are you satisfied with the performance of the system?	4.5	0.3

Table 8: Results of the subjective evaluation of the multi-domain dialog system with real users (1=worst, 5=best evaluation)

6. Conclusions

The development of multi-domain dialog systems still poses a grand challenge due mainly to two big problems: (1) identifying the domain for the current turn is not obvious and misrecognizing it would lead the system to assign the management of the current turn to a dialog manager that is not able to deal with it appropriately, (2) once the domain has been identified, managing the interaction for that domain making the most of the available single-domain resources.

In this paper we describe a proposal that addresses these two questions with an architectures that can be incorporated on top of the already existing single-domain systems in order to build a multi-domain dialog system in such a way that the single-domain system do not have to be modified and their resources (e.g. existing ASR and DM modules) can be exploited. The process performed using this architecture is comprised of two stages in which for each user turn the best single-domain dialog system is used to generate the multi-domain system response. In the first stage, the single domains are sorted from most to less probable taking into account the user utterance. At a second stage, a domain is chosen from the list using a statistical model of the conversation in each domain. Incorporating the second stage allows to refine the ranking that is computed with the first stage.

We have evaluated the proposal by building a multi-domain dialog system from three already existing systems in different but related domains. The evaluation results show that introducing the second stage outperforms the scenario in which only the first stage is used. Although the first stage has very good performance, as it is based only on the user utterance if the wording used is very general (e.g. it does not contain words that are highly specific for a particular domain), in these situations adding a second stage based on a model of the interaction in each domain make it possible to take the system back on track by selecting a more appropriate domain. The evaluation has been performed using a simulated user that allow the generate a high number of dialogs to assess the proposal in detail covering a high number of situations. These positive outcomes also apply with real recruited users. The results validate our proposal both when the multi-domain system generated is used in each of the single domains separately and when the conversation is multi-domain.

For future work we are interested in exploring new avenues of research to directly build multi-domain systems that do not rely on previously existing single-domain systems. The challenge in this case will be to maintain the cost-effectiveness of the approach by using novel learning algorithms that work over scarce multi-domain conversational data.

Acknowledgements

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

References

[1] R. Pieraccini, The Voice in the Machine: Building Computers That Understand Speech, MIT Press, 2012.

[2] T. Heinroth, W. Minker, Introducing Spoken Dialogue Systems into Intelligent Environments, Kluwer Academic Publishers, 2012.

[3] M. F. McTear, Z. Callejas, D. Griol, The Conversational Interface, Springer, 2016.

[4] F. Metze, X. Anguera, E. Barnard, M. Davel, G. Gravier, Language independent search in MediaEval's Spoken Web Search task, Computer, Speech and Language 28 (5) (2014) 1066–1082.

[5] A. Stent, S. Stenchikova, M. Marge, Reinforcement learning of dialogue strategies with hierarchical abstract machines, in: Proc. of SLT'06, 2006, pp. 210–213.

[6] M. Tsai, The VoiceXML dialog system for the e-commerce ordering service, in: Proc. of CSCWD'05, 2005, pp. 95–100.

[7] K. Reschke, A. Vogel, D. Jurafsky, Generating Recommendation Dialogs by Extracting Information from User Reviews, in: Proc. of ACL'13, 2013, pp. 499–504.

[8] K. Kopp, M. Britt, K. Millis, A. Graesser, Improving the efficiency of dialogue in tutoring, Learning and Instruction 22(5) (2012) 320–330.

[9] H. Hofmann, A. Silberstein, U. Ehrlich, A. Berton, C. Muller, A. Mahr, Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice, Springer, 2014, Ch. Development of Speech-Based In-Car HMI Concepts for Information Exchange Internet Apps, pp. 15–28.

[10] T. Misu, A. Raux, R. Gupta, I. Lane, Situated language understanding for a spoken dialog system within vehicles, Computer Speech and Language 34 (2015) 186–200.

[11] W. Minker, T. Heinroth, P. Strauss, D. Zaykovskiy, Human-Centric Interfaces for Ambient Intelligence, Elsevier, 2010, Ch. Spoken Dialogue Systems for Intelligent Environments, pp. 453–478.

[12] T. Bickmore, K. Puskar, E. Schlenk, L. Pfeifer, S. Sereika, Maintaining reality: Relational agents for antipsychotic medication adherence, Interacting with Computers 22 (2010) 276–288.

[13] O. Horchak, J.-C. Giger, M. Cabral, G. Pochwatko, From demonstration to theory in embodied language comprehension: A review, Cognitive Systems Research 29-30 (2014) 66–85.

[14] J. Planells, L. Hurtado, E. Segarra, E. Sanchis, A multi-domain dialog system to integrate heterogeneous spoken dialog systems, in: Proc. of InterSpeech'13, 2013, pp. 1891–1895.

[15] I. Lee, S. Kim, K. Kim, D. Lee, J. Choi, S. Ryu, G. G. Lee, A two-step approach for efficient domain selection in multi-domain dialog systems, Natural Interaction with Robots, Knowbots and Smartphones: Putting Spoken Dialog Systems into Practice 1 (2009) 125–133.

[16] K. Komatani, N. Kanda, M. Nakano, K. Nakadai, H. Tsujino, T. Ogata, H. G. Okuno, Multi-domain spoken dialogue system with extensibility and robustness against speech recognition errors, in: Proc. of SIGdial'06, 2006, pp. 9–17.

[17] B. Pakucs, Towards dynamic multi-domain dialogue processing, in: Proc. of Eurospeech'03, 2003.

[18] B. Lin, H. Wang, L. Lee, A distributed architecture for cooperative spoken dialogue agents with coherent dialogue state and history, in: Proc. of ASRU'99, 1999.

[19] M. Gasic, D. Kim, P. Tsiakoulis, S. Young, Distributed dialogue policies for multi-domain statistical dialogue management, in: Proc. of ICASSP'15, 2015.

[20] M. Gasic, D. Kim, P. Tsiakoulis, S. J. Young, Distributed dialogue policies for multi-domain statistical dialogue management, in: Proc. of ICASSP'15, 2015, pp. 5371–5375.

[21] S. Seneff, D. Goddeau, C. Pao, J. Polifroni, Multimodal discourse modelling in a multi-user multi-domain environment, in: Proc. ICSLP'96, 2002.

[22] W.-T. Hsu, H.-M. Wang, Y.-C. Lin, The design of a multi-domain chinese dialogue system, in: Proc. of ISCSLP'02, 2002.

[23] O. Lemon, A. Gruenstein, S. Peters, Collaborative activities and multi-tasking in dialogue systems: Towards natural dialogue with robot, TAL. Traitement automatique des langues 43 (2) (2002) 131–154.

[24] C. Lee, S. Jung, S. Kim, G. G. Lee, Example-based dialog modeling for practical multi-domain dialog system, Speech Communication 51 (2009) 466–484.

[25] J. Choi, K. Kim, S. Kim, D. Lee, I. Lee, G. G. Lee, Multi-domain spoken dialog system for information access in mobile environments, in: Proc. of ASRU'11, 2011.

[26] M. O. Dzikovska, J. F. Allen, M. D. Swift, Linking semantic and knowledge representations in a multi-domain dialogue system, Journal of Logic and Computation 18 (3) (2008) 405–430.

[27] D. Vandyke, P.-H. Su, M. Gasic, N. Mrksic, T.-H. Wen, S. Young, Multi-domain dialogue success classifiers for policy training, in: Proc. of ASRU'15, 2015.

[28] W. Ge, B. Xu, Dialogue management based on multi-domain corpus, in: Proc. of SIGdial'15, 2015, pp. 364–373.

[29] M. Nakano, S. Sato, K. Komatani, K. Matsutama, K. Funakoshi, H. Okuno, A two-stage domain selection framework for extensible multi-domain spoken dialogue systems, in: Proc. of SIGDial'11, 2011.

[30] O. Z. Khan, J.-P. Robichaud, P. Crook, R. Sarikaya, Hypotheses ranking and state tracking for a multi-domain dialog system using multiple asr alternates, in: Proc. of InterSpeech'15, 2015.

[31] M. Doulaty, O. Saz, T. Hain, Data-selective transfer learning for multi-domain speech recognition, in: Proc. of InterSpeech'15, 2015.

[32] A. Celikyilmaz, D. Hakkani-Tur, G. Tur, Multi-domain spoken language understanding with approximate inference, in: Proc. of InterSpeech'11, 2011.

[33] M. Jeong, G. G. Lee, Multi-domain spoken language understanding with transfer learning, Speech Communication 51 (5) (2009) 412–424.

[34] T.-H. Wen, M. Gasic, N. Mrksic, L. Rojas-Barahona, P.-H. Su, D. Vandyke, S. Young, Toward multi-domain language generation using recurrent neural networks, in: Proc. of NIPS Workshop on Machine Learning for Spoken Language Understanding and Interaction, 2015.

[35] R. Jiang, R. E. Banchs, S. Kim, K. H. Yeo, A. Niswar, H. Li, Web-based multimodal multi-domain spoken dialogue system, in: Proc. of IWSDS'14, 2014, pp. 182–187.

[36] C. Wootton, M. McTear, T. Anderson, Utilizing online content as domain knowledge in a multi-domain dynamic dialogue system, in: Proc. of InterSpeech'07, 2007, pp. 122–125.

[37] D. Traum, S. Larsson, Current and New Directions in Discourse and Dialogue, Kluwer, 2003, Ch. The Information State Approach to Dialogue Management, pp. 325–353.

[38] S. Larsson, S. Ericsson, Godis - issue-based dialogue management in a multi-domain, multi-language dialogue system, in: Proc. of the ACL-02 Demonstrations Session, 2002.

[39] D. Hakkani-Tur, G. Tur, L. Heck, A. Fidler, A. Celikyilmaz, A discriminative classification-based approach to information state updates for a multi-domain dialog system, in: Proc. of InterSpeech'12, 2012.

[40] S. Chu, I. O'Neill, P. Hanna, M. McTear, An approach to multistrategy dialogue management, in: Proc.of Interspeech'05, 2005, pp. 865–868.

[41] M. Walker, D. Hindle, J. Fromer, G. D. Fabbrizio, C. Mestel, Evaluating Competing Agent Strategies for a Voice Email Agent, in: Proc. of Eurospeech'97, 1997, pp. 2219–2222.

[42] M. Nakano, K. Funakoshi, Y. Hasegawa, H. Tsujino, A framework for building conversational agents based on a multi-expert model, in: Proc. of SIGdial'08, 2008, pp. 88–91.

[43] T. Hiraoka, G. Neubig, K. Yoshino, T. Toda, S. Nakamura, Active learning for example-based dialog systems, in: Proc. of IWSDS'16, 2016.

[44] S. Jung, C. Lee, K. Kim, D. Lee, G. Lee, Hybrid user intention modeling to diversify dialog simulations, Computer Speech and Language 25 (2) (2011) 307–326.

[45] C. Lee, S. Jung, K. Kim, G. G. Lee, Hybrid approach to robust dialog management using agenda and dialog examples, Computer Speech and Language 24(4) (2010) 609–631.

[46] L. Nio, S. Sakti, G. Neubig, T. Toda, S. Nakamura, Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system, IEICE Transactions on Information and Systems E97 (D(6)) (2014) 1497–1505.

[47] R. E. Banchs, H. Li, Iris: a chat-oriented dialogue system based on the vector space model, in: Proc. of ACL'12, 2012.

[48] S. Young, The Statistical Approach to the Design of Spoken Dialogue Systems, Tech. rep., CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge, UK (2002).

[49] J. Williams, The best of both worlds: Unifying conventional dialog systems and pomdps, in: Proc. of InterSpeech'08, 2009, pp. 1173–1176.

[50] D. Griol, Z. Callejas, R. López-Cózar, G. Riccardi, A domain-independent statistical methodology for dialog management in spoken dialog systems, Computer, Speech and Language 28 (3) (2014) 743–768.

[51] P. Lison, A hybrid approach to dialogue management based on probabilistic rules, Computer, Speech and Language 34 (1) (2015) 232–255.

[52] J. Schatzmann, K. Weilhammer, M. Stuttle, S. Young, A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies, Knowledge Engineering Review 21 (2) (2006) 97–126, cambridge Journals.

[53] Z. Wang, H. Chen, G. Wang, H. Tian, H. Wu, H. Wang, Policy learning for domain selection in an extensible multi-domain spoken dialogue system, in: Proc. of EMNLP'14, 2014.

[54] A. Tsilfidis, I. Mporas, J. Mourjopoulos, N. Fakotakis, Automatic speech recognition performance in different room acoustic environments with and

without dereverberation preprocessing, Computer Speech & Language 27 (1) (2013) 380–395.

[55] W.-L. Wu, R.-Z. Lu, J.-Y. Duan, H. Liu, F. Gao, Y.-Q. Chen, Spoken language understanding using weakly supervised learning, Computer Speech & Language 24 (2) (2010) 358–382.

[56] O. Lemon, Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation, Computer Speech & Language 25 (2) (2011) 210–221.

[57] T. Dutoit, An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers, 1996.

[58] G. Salton, C. Yang, On the specification of term values in automatic indexing, Journal of Documentation 29 (4) (1973) 351–372.

[59] A. Gorin, G. Riccardi, J. Wright, How may i help you?, Speech Communication 23 (1997) 113–127.

[60] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874.

[61] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (1) (2004) 101–141.

[62] D. E. Rumelhart, G. E. Hinton, R. J. Williams, PDP: Computational models of cognition and perception, I, MIT Press, 1986, Ch. Learning internal representations by error propagation, pp. 319–362.

[63] D. Griol, L. Hurtado, E. Segarra, E. Sanchis, A Statistical Approach to Spoken Dialog Systems Design and Evaluation, Speech Communication 50 (8–9) (2008) 666–682, elsevier.

[64] M. Fraser, G. Gilbert, Simulating speech systems, Computer Speech and Language 5 (1991) 81–99.

[65] S. Young, Cued standard dialogue acts, Tech. rep., Cambridge University Engineering Department, Cambridge, UK (2007).

[66] D. Griol, J. Carbo, J. Molina, Bringing context-aware access to the web through spoken interaction, Journal of Applied Intelligence 38 (4) (2013) 620–640.

[67] D. Griol, J.Carbo, J. Molina, A statistical simulation technique to develop and evaluate conversational agents, AI Communications 26 (4) (2013) 355–371.

[68] N. Mrksic, D. O'Seaghdha, B. Thomson, M. Gasic, P.-H. Su, D. Vandyke, T.-H. Wen, S. Young, Multi-domain dialog state tracking using recurrent neural networks, in: Proc. of ACL'15, 2015.

Dr. David Griol obtained his Ph.D. degree in Computer Science from the Technical University of Valencia (Spain) in 2007. He has also a B.S. in Telecommunication Science from this University. He is currently visiting lecturer at the Department of Computer Science in the Carlos III University of Madrid (Spain). He has participated in several European and Spanish projects related to natural language processing and dialog systems. His research activities at the Applied Artificial Intelligence Group are mostly related to the development of statistical methodologies for the design of spoken dialog systems. His research interests include dialog management and optimization, corpus-based methodologies, user modeling and simulation, adaptation and evaluation of spoken dialog systems and machine learning approaches.

Dr. José Manuel Molina is Full Professor at Universidad Carlos III de Madrid, Spain. He joined the Computer Science Department of the same university in 1993. Currently, he coordinates the Applied Artificial Intelligence Group (GIAA). His current research focuses on the application of soft computing techniques (NN, evolutionary computation, fuzzy logic, and multiagent systems) to radar data processing, air traffic management, and e-commerce. He (co)authored up to 50 journal papers and 200 conference papers. He received a degree in telecommunications engineering from the Technical University of Madrid in 1993 and the Ph.D. degree from the same university in 1997.