A DATA-DRIVEN APPROACH TO SPOKEN DIALOG SEGMENTATION

ACCEPTED MANUSCRIPT

David Griol, José Manuel Molina, Araceli Sanchis, Zoraida Callejas

Software Engineering Department, University of Granada, Spain
{dgriol, zoraida}@ugr.es
Universidad Carlos III de Madrid, Computer Science Dept., Avda. de la Universidad 30, 28911 Leganés, Spain
molina@ia.uc3m.es, masm@inf.uc3m.es

This is the accepted version of the paper: David Griol, José Manuel Molina, Araceli Sanchis, Zoraida Callejas: A data-driven approach to spoken dialog segmentation. Neurocomputing 391: 292-304 (2020) https://doi.org/10.1016/j.neucom.2019.02.072 (https://www.sciencedirect.com/science/article/pii/S0925231219316078)

A data-driven approach to dialog structure modeling

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

[a]*Dept. of Computer Science*
Carlos III University of Madrid, Spain.
{david.griol,araceli.sanchis,josemanuel.molina}@uc3m.es
[b]*Dept. of Languages and Computer Systems,*
University of Granada, Spain.
zoraida@ugr.es

Abstract

With the advances in Language Technologies and Natural Language Processing, conversational interfaces have begun to play an increasingly important role in the design of human-machine interaction systems in a number of devices and intelligent environments. In this paper, we present a statistical model for spoken dialog segmentation and labeling based on a generative model learned using decision trees. We have applied our proposal in a practical conversational system that helps solving simple and routine software and hardware repairing problems. The results of the evaluation show that automatic segmentation of spoken dialogs is very effective for human-machine dialogs. The same statistical model has been applied to human-human conversations and provides a good baseline as well insights in the model limitation.

Keywords: Domain Knowledge Acquisition, Dialog Structure Annotation, Conversational Interfaces, Human-machine Interaction, Spoken Interaction.

1. Introduction

Speech and natural language technologies allow users to communicate in a flexible and efficient manner, making possible to access applications in which traditional input interfaces cannot be used (e.g. in-car applications, access for disabled persons, etc) [1]. Also speech-based interfaces work seamlessly with small devices (e.g., smarthphones and tablets PCs) and allow users to easily interact

Preprint submitted to Neurocomputing *July 9, 2018*

with robotic agents (e.g., Jibo[1] and Pepper[2]), invoke local applications or access remote information by means of enhanced devices and advanced conversational interfaces (e.g., Amazon Echo[3] and Google Home[4]).

In the past decade, the computational linguistics community has focused on developing language processing algorithms that can leverage the vast quantities of dialog corpus data that are generated every day. In this field, a machine learning technique could potentially reduce human effort in the knowledge engineering process and development of a new conversational system.

Dialog segmentation can be defined as the process of dividing up a dialog by one of several related notions (speaker's intention, topic flow, coherence structure, cohesive devices, etc.), identifying boundaries where the discourse changes taken into account such as specific criteria. This detection is usually based on combining different kinds of features, such as semantic similarities, inter-sentence similarities, entity repetition, word frequency, linguistic features, and prosodic and acoustic characteristics.

In this paper, we describe a machine learning approach for the automatic segmentation of spoken dialogs. The objective is to detect sequences of turns that accomplish a specific objective (tasks and subtasks) inside the dialog flow. These parts can be necessary for obtaining the final goal of the dialog, and also general parts not strictly related to the domain of the dialog system (greetings, error recovery, etc.). The detection of the dialog structure is useful to develop dynamic and user-adapted conversational interfaces. Modeling subdialog structures is also useful to extend these interfaces to deal with more complex tasks.

Our methodology for dialog segmentation decides the current phase of the dialog by means of a classification process that considers the complete history of the dialog, which is one of the main advantages regarding the previously described statistical methodologies. Another main characteristic is the inclusion of a data structure that stores the information provided by the user. The main objective of this structure is to easily encode the complete information related to the task provided by the user during the dialog history.

The remainder of the paper is as follows. Section 2 reviews related approaches to dialog structure modeling and discourse segmentation. Section 3 presents our statistical proposal for automatic segmentation of spoken dialogs. Section 4 describes the practical application of this proposal for a conversational system acting as a customer support service for simple and routine software/hardware repairing

[1]https://www.jibo.com/
[2]https://www.ald.softbankrobotics.com/en/cool-robots/pepper
[3]https://www.amazon.com/echo-superbowl-commercial/
[4]https://madeby.google.com/home/

problems. In Section 5 we discuss the evaluation results of the application of our proposal for this system. Finally, in Section 6 we present the conclusions and outline guidelines for future work.

2. Related work

Research on data-driven approaches to dialog structure modeling is relatively new and focuses mainly on recognizing a structure of a dialog as it progresses [2, 3]. In the literature, there are different methodologies for discourse segmentation and the construction of dialog models including task/subtask information. Unsupervised clustering and segmentation techniques are used in [4] to identify concepts and subtasks in task-oriented dialogs.

Passoneau and Litman [5] presented an algorithm for identifying topic boundaries that uses decision trees to combine multiple linguistic features extracted from corpora of spoken text. These include prosodic features such as pause duration, lexical features such as the presence of certain cue-phrases near boundary candidates, and deeper semantics questions such as whether two noun phrases on opposite sides of a boundary candidate.

Yamron [6] presented an approach to segmentation that models an unbroken text stream as an unlabeled sequence of topics using Hidden Markov Models. Ponte presented in [7] an approach based on information retrieval methods that map a query text into semantically related words and phrases. The approach presented in [8] is based on using adaptive language models and cue-word features for incrementally building an exponential model to extract features that are correlated with the presence of boundaries in labeled training text.

There is also a wide range of natural language processing applications for which discourse segmentation assists in. For instance, Angheluta, Busser and Moens adapted a three-step segmentation algorithm for automatic text summarization [9]. Their algorithm uses generic topical cues for detecting the thematic structure of a text using synonymy to reduce the vocabulary.

Different proposals apply discourse segmentation to segment text into different fragments in a preprocessing phase of information retrieval and question-answering systems to improve their operation [10, 11]. Walker applies this kind of techniques for anaphora resolution[12]. Finally, different studies show the benefits of using discourse segmentation for question answering tasks in order to take into account the context for the interpretation and answer questions [13], and also for dialog acts segmentation and classification [14].

3. Our proposed methodology for spoken dialog segmentation

We represent dialogs as a sequence of pairs (A_i, U_i), where A_i is the output of the dialog manager (the system answer) 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 [15]. Each dialog is represented by:

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

where A_1 is the greeting turn of the system, and U_n is the last user turn. We refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i.

We consider a task-oriented dialog to be the result of incremental creation of a shared plan by the participants [16]. This shared plan consists of several tasks and subtasks. The goal of subtask segmentation is to predict if the current utterance in the dialog is part of the current subtask or it starts a new subtask. We model this prediction problem as a maximization task as the following equation shows:

$$\hat{S}_i = \underset{S_i \in \mathcal{S}}{\operatorname{argmax}} P(S_i | U_1 \cdots U_{i-1}, S_1 \cdots S_{i-1}) \tag{1}$$

where set $\mathcal{S}$ contains all the possible kinds of tasks/subtasks defined for the dialog segmentation and U_n is the semantic representation of the user utterance at time n in terms of the list of features provided by the Spoken Language Understanding (SLU) module of the conversational agent. The prediction of the current task, that is a local process, takes into account the previous history of the dialog, that is to say, the sequence of user turns and dialog segments preceding time i.

The lexical, syntactic and semantic information associated to the speaker u's ith turn (U_i) is usually represented by means of different information sources:

- the words uttered;
- dialog acts, which represent the meaning of an utterance at the level of the speaker's intention in producing that utterance (e.g., *Acceptance*, *Not-Understood*, *Tag-Question*, or *Apology*).
- part of speech tags, also called word classes or lexical categories. Common linguistic categories include noun, adjective, and verb, among others;
- predicate-argument structures, used by SLU modules in various contexts to represent relations within a sentence structure. They are usually represented as triples (subject-verb-object).

- named entities: sequences of words that refer to a unique identifier. This identifier may be a proper name (e.g., organization, person or location names), a time identifier (e.g., dates, time expressions or durations), or quantities and numerical expressions (e.g., monetary values, phone numbers, etc.).

The main problem to resolve Equation 1 is regarding the number of possible sequences of user's utterances preceding the current one, which could be very large in a practical conversational system. To solve this problem, we define a data structure, which we call *User Register* (UR), and contains the information provided by the user throughout the previous history of the dialog. The prediction of the current phase of the dialog S_i is then modeled by means of the following equation:

$$\hat{S}_i = \underset{S_i \in \mathcal{S}}{\operatorname{argmax}}\, P(S_i | UR_{i-1}, S_1 \cdots S_{i-1}) \tag{2}$$

As a practical implementation of this equation, we propose the use of a classification process that takes the semantic information of the user's utterances and the sequence of previous dialog tasks as input, and provides the probability of the dialog being at each of the dialog tasks as output. The C4.5 decision tree learning algorithm has been used to learn this classification model, using the Weka machine learning software for classifying the complete list of features contained in the user register. Using this model, the current segment of the dialog is selected by taking into account the previous list of dialog segments detected and the complete list of features provided by the SLU module.

4. Practical tasks

We have used two corpora for the evaluation of our proposal: LegAssistant and Let's Go!, which characteristics are summarized in Table 1 and detailed in the following subsections.

4.1. LegAssistant corpus

The LegaAssistant corpus is related to the problem solving domain of a practical spoken dialog system, which acts as a customer support service to help solving simple and routine software/hardware repairing problems, both at the domestic and professional levels.

The definition of the system's functionalities and dialog strategy was carried out by means of the analysis of 300 human-human conversations (*Human-Human dialogs*, HH) provided by real assistants attending the calls of users with a software/hardware problem at the City Council of Leganés (Madrid, Spain).

Table 1: Main characteristics of the corpora.

	LegaAssistant	Let's Go!	DSTC
Number of dialogs	150	235	10,415
Number of user turns	4,002	4,222	122,025
Average number of user turns per dialog	5.6	18.0	11.7
Number of dimensions of the input feature vector	19	11	13
Number of possible system acts	29	26	17

The labeling defined for this corpus contains different types of information, that have been annotated using a multilevel approach similar to the one proposed in the Luna Project [17]. The first levels include segmentation of the corpus in dialog turns, transcription of the speech signal, and syntactic preprocessing with POS-tagging and shallow parsing. The next level consists of the annotation of main information using attribute-value pairs. The other levels of the annotation show contextual aspects of the semantic interpretation. These levels include the predicate structure, the relations between referring expressions, and the annotation of dialog acts.

The attribute-value annotation uses a predefined domain ontology to specify concepts and their relations. The attributes defined for the task include *Concept*, *Computer-Hardware*, *Action*, *Person-Name*, *Location*, *Code*, *TelephoneNumber*, *Problem*, etc.

Dialog act (DA) annotation was performed manually by three annotators on speech transcriptions previously segmented into turns. The DAs defined to label the corpus can be classified into the following categories: i) Core DAs: *Action-request*, *Yes-answer*, *No-answer*, *Answer*, *Offer*, *ReportOnAction*, *Inform*; ii) Conventional DAs: *Greet*, *Quit*, *Apology*, *Thank*; iii) Feedback-Turn management DAs: *ClarificationRequest*, *Ack*, *Filler*; iv) Non interpretable DAs: *Other*.

The original FrameNet[5] description of frame elements was adopted for the predicate-argument structure annotation, introducing new frames and roles related to hardware/software only in case of gaps in the FrameNet ontology. Some of the frames included in this representation are *Telling*, *Greeting*, *Contacting*, *Statement*, *Recording*, *Communication*, *Being operational*, *Change operational state*, etc. Table 2 shows the complete set of features used for the labeling of the different

[5]https://framenet.icsi.berkeley.edu/fndrupal/

corpora and for the experiments that are presented in this paper.

Dialog Acts	**Core dialog acts** (*Info-request*, *Action-request Yes-answer*, *No-answer*, *Answer*, *Offer*, *ReportOnAction*, *Inform*), **Conventional dialog acts** (*Greet*, *Quit*, *Apology*, *Thank*), **Feedback/Turn management dialog acts** (*ClarificationRequest*, *Ack*, *Filler*), **Non interpretable/Non classifiable** (*Other*)
Attribute-value	Concept, Computer-Hardware, Time-PartoftheDay, Negation, Action, Person-Name, Person-Surname, Location-Institution, Code, Location-Other, Location-TelephoneNumber, Ordinal-Number, Cardinal-Number, Time-RelativeTime, Problem, Person-Position
Predicate information	Telling, Greeting, Contacting, Statement, Recording, Communication. Being operational, Change operational state, Operational testing, Being in operation.

Table 2: List of semantic features used for the labeling of the corpus

The basic structure of the dialogs is usually composed by the sequence of the following tasks: *Opening*, *Problem-statement*, *User-identification*, *Problem-clarification*, *Problem-resolution*, and *Closing*. This set of tasks contains a list of subtasks, such as *Problem-description*, *Problem-Request*, *Problem-Confirmation*, *Brand-Identification*, *Model-Identification*, *Help-Request*, *Message-Confirmation*, *Name-Identification*, *Resolution-Confirmation*, etc. The shared plan is represented as a data register that encapsulates the task structure, dialog act structure, attribute-values and predicate-argument structure of utterances. Figure 1 shows an example of the incremental evolution of dialog structure with the complete set of tasks and substasks. It can be observed the difficulty of correctly detecting the complete structure of the dialog.

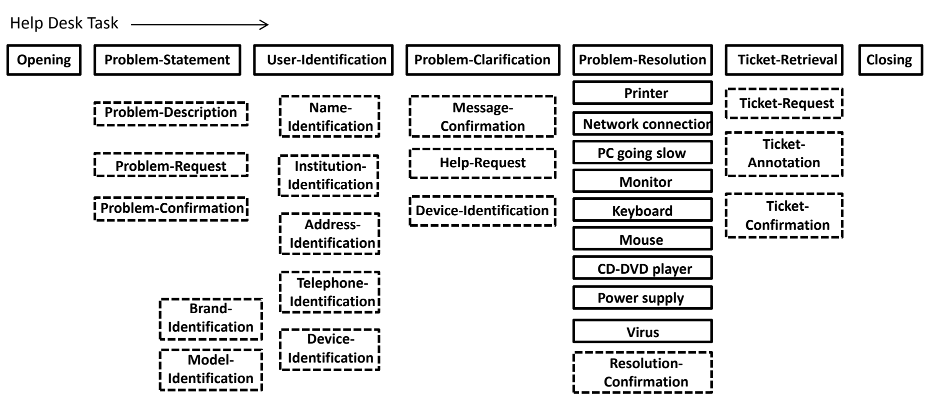

Figure 1: Incremental evolution of the dialog structure

During the *Problem-statement* task, the caller explains the problem the reasons why he/she calls the help-desk. In the *User-identification* task, the operator asks for additional information regarding the identity of the caller. Once the caller has described the problem, the operator can ask for additional information to clarify it during the *Problem-clarification* task.

During the *Problem-resolution* task, the operator asks the user to perform specific tests. We have defined nine different subtasks inside this generic segment, given that our goal is to detect not only that the dialog is in this segment, but also what are the specific problem that has to be resolved: *Printer* (P4), *Network connection* (P5), *PC going slow* (P6), *Monitor* (P7), *Keyboard* (P8), *Mouse* (P9), *CD-DVD player* (P10), *Power supply* (P11), and *Virus* (P12). The operator assigns a ticket number for the current call if the problem has not been solved after this task (P13). The user must take note of this number and inform about this to the operator. The dialog ends at the *Closing* phase, in which the operator also tries to give a useful advice related to the described problem (P14).

S1: Hi. How may I help you?
U1: Hello, my printer does not print correctly.
S2: Tell me your name and surname.
U2: My name is Ricardo Santamaria.
S3: Can you tell me your phone extension?
U3: Nine four sixty-three.
S4: Can you tell me the brand and model?
U4: HP Officejet Pro 8610.
S5: Is the printer on now?
U5: It is on.
S6: Can you tell me the message on the printer display?
U6: Paper jam is written.
S7: Can you see the active printer from the control panel of your PC?
U7: It is active.
S8: Please try the following steps. Say next after each step. Turn the printer off, and then on again.
U8: Next.
S9: Print a Printer Status Report.
U9: Next.
S10: Has this process solved your problem?
U10: No.
S11: Remove any loose paper from the main and optional trays.
U11: Next.
S12: Clear any jammed paper from the rear of the printer.
U12: Next.
S13: Has this process solved your problem?
U13: Yes.
S14: Anything else?
U14: No, thank you.
S15: Most of these steps will also work for troubleshooting an office copier. Have a nice day!

Figure 2: Example of a Human-Machine dialog (translation from Spanish to English)

4.2. Let's Go corpus

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

The system has had many users since it was made available for the general public in 2005, with more than 20,000 calls collected just from March to December of 2005 [18], so there is a substantial dataset that can be used to train a dialog model. In addition, this large amount of data from spoken interactions has been acquired with real callers, rather than lab testers. The system integrates the RavenClaw dialog manager [19], the Sphinx 2 speech recognition engine[6] and a domain-specific voice built with the Festival/Festvox toolkit and deployed on the Cepstral Swift engine[7].

In a study carried out after the first two operative years of the system [20], the average daily call traffic for the past year oscillated between 40 and 60. The average length of dialogs was 14 turns. However the distribution of dialog turn lengths was bi-modal, with a first peak at 0 turns (10% of the dialogs) and a second one around 10 turns. Complete dialogs have a 79% dialog success rate. Typical failures detected in the dialogs included system wrongly interrupting its turn (52.0% of calls), system failing to take a turn (47.1% of calls), system failing to yield a turn on user barge-in (43.1% of calls), system taking an extra turn (39.2% of calls), and system wrongly barging in on user (15.7% of calls). At the end of April 2015, the number of calls is typically averaging 40 calls during weekdays and 100 calls during weekends. The current version of the dialog system provides an estimated dialog success of 75.11% with an average number of 12.24 turns, with a relatively large standard deviation of 11.5 turns [21].

Figure 3 shows an example of a dialog extracted from the Let's Go corpus [22]. Each call to the system starts with a welcome message that prompts the user to make a request. Then, the system waits for the user's response and grabs concepts such as question type (e.g., *When is the next bus to X?*, *How can I go from X to Y?*) or departure and arrival times and places. To be successful, calls require three or four pieces of information from the user: a departure stop, a destination, a travel time, and, optionally a bus route. Stops can be specified in one of three ways: the nearest intersection to the stop (e.g., Forbes [Avenue] at Murray [Avenue]), a neighborhood (e.g., Oakland), or a landmark or other point of interest (Pittsburgh

[6]http://cmusphinx.sourceforge.net
[7]http://www.cepstral.com

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

International Airport, Waterworks Mall). The system explicitly prompts the user to provide the missing information to complete the query. Once the system has the required information to answer the user's query, it submits a query to the database, presents the results to the user, and prompts for a new query.

We have chosen the Let's Go task to evaluate our proposal for several reasons. Firstly, the corpus available was gathered from a real task in an operative dialog system that provided its service to real users. This poses a challenge to build realistic user models and find new dialog strategies that are at least as good as the hand-crafted system. Secondly, Let's Go is a common ground for experimentation and evaluation within the dialog system community, which therefore makes our results directly comparable to the alternatives presented by other authors, and this is why it has been intensively used by researchers in the last years [23, 24, 25, 22].

The Dialog State Tracking Challenge (DSTC)[8] is a regularly held community challenge which provides annotated data with dialog state information. The first challenge was held in 2013 [**?**]), while the most recent one (DSTC6) was organized in 2017. A corpus of 10,415 dialogs with 122,025 dialog acts was distributed among the scientific community as a common testbed for the first challenge.

With regard the semantic representation defined for the task, the system uses

[8]`https://www.microsoft.com/en-us/research/event/dialog-state-tracking-challenge/`

a set of user dialog acts that has been classified into 16 categories following the criteria described in [26]. A total of 16 categories of user dialog acts were defined. Four of the dialog acts are used to model where the user is leaving from (monument, pair of road names, neighborhood, or stop). The four dialog acts used for modeling the place of arrival are similar. Six dialog acts are used for describing the user's required time of travel (next bus or specific times). The *meth* node describes whether the user is asking for a bus with some constraints, is finished or wants to restart. The dialog act *disc* models how the user issues "discourse" actions, which relate to only one turn in a dialog.

A total of 36 system dialog acts were defined. These dialog acts can also be classified into 5 groups: *formal* (dialog formalities like "welcome"), *results* (presentation of search results), *queries* (request for values to fill slots), *statusreports* (when the system reports about its status, e.g. "looking up database"), *error* (error messages), and *instructions* (instructions to the user how to speak to the system).

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

5. Results and discussion

The developed methodology for the task detection has been evaluated by means of the HM and HH dialogs of the described corpora. Figure 4 shows the distribution of dialog segments annotated in each corpus. As can be seen, the tasks distribution is very different, presenting HH dialogs an additional 27.31% percentage of situations that has been labeled as *Out of the Task* (P15).

As HH dialogs are spontaneous, they present several differences with regard to the HM dialogs. The main one is the great difference in the average number of turns (16.18 turns in the HM corpus and 25.71 for the HH dialogs). This is because HH dialogs present other minor topics (like small talks about other persons, previous problems, holidays, etc), a high frequency of interruptions, cut-off phrases, and overlapped contributions. This makes that the 18.24% of the utterances of the HH corpus have been labeled as *Out of the Task*.

Analyzing the annotation available for the DA level, we measured that in average an HH dialog is composed of 37.9$\pm$7.3 (Std. Dev.) DAs, whereas a HM

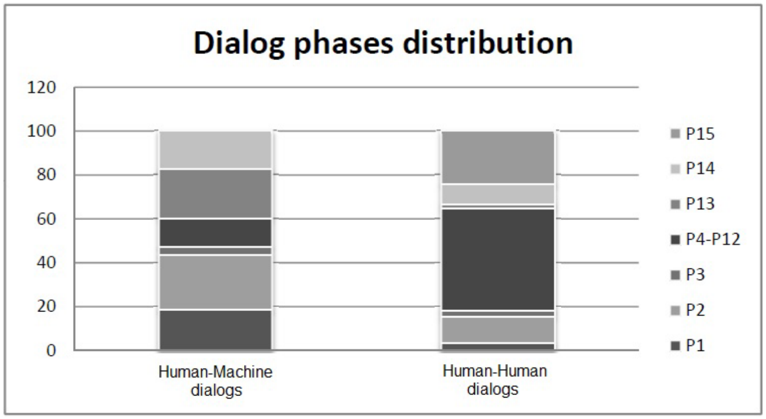

Figure 4: Distribution of dialog segments annotated in the HM and HH dialogs

dialog is composed of 21.9±5.4. The difference between average lengths shows how HH spontaneous speech can be redundant, while HM dialogs are more limited to an exchange of essential information. The standard deviation of a conversation in terms of DAs is considerably higher in the HH corpus than in the HM ones. This can be explained by the fact that the HM dialogs follow a unique, previously defined task-solving strategy that does not allow digressions.

The evaluation of the statistical dialog segmentation technique was carried out *turn by turn* using a five-fold cross validation process. Each one of the two corpus was randomly split into five subsets. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80% of the dialogs was used as the training set. Table 3 shows the results of the application of this methodology for the HM and HH corpora. The results show how the prediction is improved once the different SLU features are incorporated to the model. As can be seen, the proposed methodology successfully adapts to the requirements of the HM dialogs, since a 0.98 F-measure is obtained, measuring the dialog segments provided by the developed module that are equal to the segment annotated in the corpus for the HM dialogs. This value is reduced to 0.79 for the HH dialogs, since the *Out of the Task* class is usually confused with the rest of dialog segments related to the task. Therefore, the methodology adapts to the very different nature that has been described for both kind of dialogs.

Finally, we learned a model with the total 150 HM dialogs and evaluated it using the total 150 HH dialogs. This experimentation was designed to evaluate if a model learned with HM dialogs can detect the task-related structure of spontaneous HH conversations. The main challenge of this experiment is that only a maximum of 81.76% can be achieved due to the 18.24% *Out of the task* that is only present in the HH corpus. As can be observed, the model successfully adapts to detect the task-related parts in the HH dialogs, achieving a 0.67 F-measure.

	Precision	Recall	F-measure
HM corpus for learning and evaluating			
Attribute-Values	0.89	0.87	0.88
DAs + Attribute-Values	0.94	0.92	0.93
Complete set	0.97	0.98	0.98
HH corpus for learning and evaluating			
Attribute-Values	0.72	0.60	0.66
DAs + Attribute-Values	0.86	0.71	0.78
Complete set	0.87	0.72	0.79
HM corpus for learning - HH corpus for evaluating			
Attribute-Values	0.62	0.55	0.58
DAs + Attribute-Values	0.69	0.61	0.65
Complete set	0.73	0.63	0.67

Table 3: Average results of the evaluation of the proposed dialog segmentation technique for the help desk system

6. Conclusions and future work

In this paper, we have presented a statistical approach for automatically dialog segmentation in conversational interfaces. This approach uses feature selection to collect a set of informative features into a model that includes both the information provided by the user and the system prompts. This model can be used to predict where boundaries occur in the dialog, helping the dialog manager in the selection of the next system prompt. The results of the evaluation of this methodology to develop a dialog segmentation module for a help desk conversational system show that the statistical approach successfully adapts to the requirements of the task, not only separately for human-machine and human-human dialogs acquired for this task, but also it is possible to successfully detect the task-related information that is present in spontaneous human-human dialogs by learning a model only with human-machine dialogs. As a future work, we want to perform a more detailed analysis of the situations that have been labeled as *Out of the Task*, studying if our proposal is able to differentiate these situations. We also want to consider the incorporation of additional information regarding the user, such as specific user profiles adapted to the each specific interaction domain. Finally, we want also to apply our proposal to more complex dialog domains.

Acknowledgements

This work was supported in part by Projects TRA2015-63708-R and TRA2016-78886-C3-1-R.

Acknowledgements

Work partially supported by Projects TRA2015-63708-R and TRA2016-78886-C3-1-R.

References

[1] M. McTear, Z. Callejas, D. Griol, The Conversational Interface: Talking to Smart Devices, Springer, 2016.

[2] T. Hiraoka, G. Neubig, S. Sakti, T. Toda, S. Nakamura, Learning cooperative persuasive dialogue policies using framing, Speech Communication 84 (2016) 83–96.

[3] U. Guz, G. Tur, D. Hakkani-Tur, S. Cuendet, Cascaded model adaptation for dialog act segmentation and tagging, Computer Speech & Language 24 (2) (2010) 289–306.

[4] A. Chotimongkol, Learning the structure of task-oriented conversations from the corpus of in-domain dialogs, Ph.D. thesis, CMU, Pittsburgh (USA) (2008).

[5] R. Passoneau, D. Litman, Discourse segmentation by human and automated means, Computational Linguistics 23 (1997) 103–139.

[6] J. Yamron, Topic detection and tracking segmentation task, in: Proc. Broadcast News Transcription and Understanding Workshop, 1998.

[7] J. Ponte, W. Croft, Text segmentation by topic, in: Proc. of ECDL, 1997, pp. 120–129.

[8] D. Beeferman, A. Berger, J. Lafferty, Statistical models for text segmentation, in: Machine Learning, Vol. 34, 1999, pp. 177–210.

[9] R. Angheluta, R. D. Busser, M. Moens, The use of topic segmentation for automatic summarization, in: Proc. of ACL, 2002, pp. 66–70.

[10] M. Kaszkiel, J. Zobel, Passage retrieval revisited, in: Proc. of ACM SIGIR, 1997, pp. 178–185.

[11] A. Moffat, R. Sacks-davis, R. Wilkinson, J. Zobel, Efficient retrieval of partial documents, in: Information Processing and Management, Vol. 31, 1995, pp. 361–377.

[12] M. A. Walker, Centering, anaphora resolution, and discourse structure, Oxford University Press, 1998, pp. 401–435.

[13] J. Chai, R. Jin, Discourse structure for context question answering, in: Proc. of HLT-NAACL, 2004, pp. 23–30.

[14] J. Ang, Y. Liu, E. Shriberg, Automatic dialog act segmentation and classification in multiparty meetings, in: Proc. of ICASSP, 2005, pp. 1061–1064.

[15] D. Griol, Z. Callejas, R. Lï¿½pez-Cï¿½zar, G. Riccardi, A domain-independent statistical methodology for dialog management in spoken dialog systems, Computer, Speech and Language 28 (3) (2014) 743–768.

[16] S. Bangalore, G. DiFabbrizio, A. Stent, Learning the Structure of Task-Driven Human-Human Dialogs, IEEE Transactions on Audio, Speech, and Language Processing 16(7) (2008) 1249–1259.

[17] E. Stepanov, G. Riccardi, A. Bayer, The Development of the Multilingual LUNA Corpus for Spoken Language System Porting, in: Proc. of LREC, 2014, pp. 2675–2678.

[18] A. Raux, B. Langner, A. Black, M. Eskenazi, Let's go public! taking a spoken dialog system to the real world, in: Proc. of Interspeech'05, Lisbon, Portugal, 2005, pp. 885–888.

[19] D. Bohus, A. Rudnicky, The RavenClaw dialog management framework: Architecture and systems, Computer Speech and Language 23 (3) (2009) 332–361.

[20] A. Raux, B. Langner, A. Black, M. Eskenazi, Doing research on a deployed spoken dialogue system: One year of Lets Go! experience, in: Proc. of Interspeech'06, Pittsburgh, Pennsylvania, USA, 2006, pp. 65–68.

[21] J. Lopes, M. Eskenazi, I. Trancoso, From rule-based to data-driven lexical entrainment models in spoken dialog systems, Computer Speech and Language 31 (2015) 87–112.

[22] H. Hastie, N. Merigaud, X. Liu, O. Lemon, "Let's Go, DUDE!" Using the Spoken Dialogue Challenge to teach Spoken Dialogue development, in: Proc. of SLT'10, Berkeley, USA, 2010, pp. 466–471.

[23] A. Schmitt, S. Ultes, W. Minker, A Parameterized and Annotated Spoken Dialog Corpus of the CMU Let's Go Bus Information System, in: Proc. of LREC'12, Istanbul, Turkey, 2012, pp. 3369–3375.

[24] A. Black, S. Burger, B. Langner, G. Parent, M. Eskenazi, Spoken dialog challenge 2010, in: Proc. of SLT'10, Berkeley, USA, 2010, pp. 448–453.

[25] J. Williams, I. Arizmendi, A. Conkie, Demonstration of AT&T Let's Go: A production-grade statistical spoken dialog system, in: Proc. of SLT'10, Berkeley, California, USA, 2010, pp. 157–158.

[26] B. Thomson, K. Yu, S. Keizer, M. Gasic, F. Jurcicek, F. Mairesse, S. Young, Bayesian dialogue system for the Let's Go Spoken Dialogue Challenge, in: Proc. of SLT'10, Berkeley, USA, 2010, pp. 460–465.