INTERSPEECH 2009 BRIGHTON

A Statistical Dialog Manager for the LUNA Project

David Griol[1], *Giuseppe Riccardi*[2], *Emilio Sanchis*[3]

[1]Dept. of Computer Science, Universidad Carlos III de Madrid, Leganés (Spain)
[2]Dept. of Information Engineering and Computer Science, University of Trento, Povo (Italy)
[3]Dept. de Sistemes Informatics i Computació, Universitat Politècnica de València, València (Spain)

`dgriol@inf.uc3m.es, riccardi@disi.unitn.it, esanchis@dsic.upv.es`

10.21437/Interspeech.2009-93

Abstract

In this paper, we present an approach for the development of a statistical dialog manager, in which the system response is selected by means of a classification process which considers all the previous history of the dialog to select the next system response. In particular, we use decision trees for its implementation. The statistical model is automatically learned from training data which are labeled in terms of different SLU features. This methodology has been applied to develop a dialog manager within the framework of the European LUNA project, whose main goal is the creation of a robust natural spoken language understanding system. We present an evaluation of this approach for both human machine and human-human conversations acquired in this project. We demonstrate that a statistical dialog manager developed with the proposed technique and learned from a corpus of human-machine dialogs can successfully infer the task-related topics present in spontaneous human-human dialogs.

Index Terms: Spoken Dialog Systems, Dialog Management, Statistical Methodologies, Classification Techniques.

1. Introduction

Learning statistical approaches to model the different modules that compose a dialog system has been of growing interest during the last decade [1]. Models of this kind have been widely used for speech recognition and also for language understanding. Even though in the literature there are models for dialog managers that are manually designed, over the last few years, approaches using statistical models to represent the behavior of the dialog manager have also been developed [2], [3], [4], [5]. These approaches are usually based on modeling the different processes probabilistically and learning the parameters of the different statistical models from a dialog corpus.

Continuous advances in the field of spoken dialog systems make the processes of design, implementation and evaluation of dialog management strategies more and more complex. The motivations for automating dialog learning are focused on the time-consuming process that hand-crafted design involves and the ever-increasing problem of dialog complexity. Statistical models can be trained from real dialogs, modeling the variability in user behaviors. Although the construction and parameterization of the model depends on the expert knowledge of the task, the final objective is to develop dialog systems that have a more robust behavior, better portability, and are easier to adapt to different user profiles or tasks.

Recently, we have presented a statistical approach for the construction of a dialog manager [6]. The dialog managers created following this proposal, are mainly based on the modelization of the sequences of the system and user spoken language understanding (SLU) features and the introduction of a partition in the space of all the possible sequences of these features. This partition, which is defined taking into account the data supplied by the user throughout the dialog, makes the estimation of a statistical model from the training data manageable.

Our dialog manager will be integrated in a dialog system developed within the framework of the LUNA project [7]. The main objective of the project is to advance the state of the art in understanding conversational speech in spoken dialog systems. Three aspects of SLU are of particular concern in LUNA: generation of semantic concept tags, semantic composition into conceptual structures, and context sensitive validation using information provided by the dialog manager. In order to train and evaluate SLU models, different corpora of spoken dialogs in multiple domains and multiple languages have been acquired. The ambitious goal of the project is to position itself at the forefront of the third generation of spoken language interfaces.

2. The Italian LUNA corpora

The LUNA corpora is currently being annotated, with the aim to collect 8,100 human-machine dialogs (HM) and 1,000 human-human dialogs (HH) in Polish, Italian and French. The Italian LUNA corpora will contain 1,000 equally partitioned HH and HM dialogs. These are recorded by CSI, an Italian customer care and technical support center. HH dialogs refer to real user conversations with CSI agents engaged in a problem solving task in the domain of software/hardware repairing. HM dialogs are acquired with a Wizard of Oz approach (WoZ). Ten different dialog scenarios inspired from the services provided by CSI were designed for the WoZ acquisition. Two corpora of 200 labeled dialogs (200 HH and 200 HM dialogs) have been used for the experiments shown in this paper.

The above data is organized in transcriptions and annotations of speech based on a new multi-level protocol studied specifically within the LUNA project, i.e. the annotation levels of words, turns, dialog acts, attribute-values, predicate argument structures. To the best of our knowledge this is the first SDS corpus (HM and HH dialogs) annotated with a multilayer approach to the annotation of lexical, semantic and dialog features. This allowed us to investigate statistical relations between the language processing layers such as shallow semantics and dialog strategies used by humans or machines.

2.1. Annotation of the Italian LUNA corpus

As stated above, the labeling defined for the LUNA corpus contains different types of information. The first levels are necessary to prepare the corpus for subsequent semantic annotation, and 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.

Dialog act (DA) annotation was performed manually by one annotator on speech transcriptions previously segmented into turns as mentioned above. The annotation unit for DAs was the utterance; however, utterances are complex semantic entities that do not necessarily correspond to turns. Hence, a segmentation of the dialog transcription into utterances was performed by the annotator before DA labeling. Figure 1 shows the list of DAs defined to label the corpus.

Core DAs	*Info-request*, *Action-request Yes-answer*, *No-answer*, *Answer*, *Offer*, *ReportOnAction*, *Inform*
Conventional DAs	*Greet*, *Quit*, *Apology*, *Thank*
Feedback/Turn management DAs	*ClarificationRequest*, *Ack*, *Filler*
Non interpretable DAs	(*Other*)

Figure 1: DA annotation defined for the LUNA corpora

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.

For the predicate-argument structure annotation, we adopted the original FrameNet description of frames and frame elements, introducing new frames and roles only in case of gaps in the FrameNet ontology. In particular, we introduced 20 new frames out of the 174 taken from FrameNet because the original definition of frames related to hardware/software, data-handling and customer assistance was too coarse-grained. In this model, the meaning of predicates (or lexical units, usually verbs, nouns, or adjectives) is conveyed by frames, conceptual structures describing prototypical situations or events and the involved participants. Some of the frames included in this representation are *Telling*, *Greeting*, *Contacting*, *Statement*, *Recording*, *Communication*, *Being operational*, *Change operational state*, *Operational testing*, *Successful action*, etc.

An example of the attribute-value, DA and predicate structure annotations of a user utterance is shown below:

Good morning, I have a problem with my mouse.
Attributes-values: *Concept*:problem; *Hardware*:mouse;
Dialog acts: *Answer*;
Predicate structure: (*Greeting*)(*Problem_description*) *Device Problem*

The system prompts defined for the WoZ acquisition have been classified into 36 different categories taking into account the following labeling: i) Task-independent prompts (*Acceptance*, *Negation*, *Not-Understood*, *Opening*, and *Closing*); ii) Prompts used to inform the user about a specified test to solve a specific problem. Nine different problems have been defined for the task : *Printer*, *Network connection*, *PC going slow*, *Monitor*, *Keyboard*, *Mouse*, *Virus*, *CD-DVD player*, and *Power-Supply*; iii) Prompts defined to require that are needed to identify the user/machine: *Name*, *Organization*, *Telephone*, *Machine-Code Brand-Model* and *Address*; iv) Prompts used for the confirmation of the problems and the identification attributes: *Confirmation-Printer*, ..., *Confirmation-Address*; v) Prompts to provide the ticket number that identifies the user call(*Ticket-Retrieval*)

An additional category called *Out of the Task* has been defined for the labeling of the prompts in the HH corpus that are not relative to the LUNA task, as it is explained in the following section.

2.2. Human-Machine and Human-Human dialogs

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 (11.18 turns in the HM corpus and 38.71 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 27.31% 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 48.9±17.4 (Std. Dev.) DAs, whereas a HM dialog is composed of 18.9±4.4. 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 WoZ follows a unique, previously defined task-solving strategy that does not allow digressions.

From a comparative analysis of the DAs occurring in the HM and HH corpora, we noticed several important differences: i) *info-request* is by far the most common DA in HM, whereas in HH *ack* and *info* share the top ranking position; ii) the most frequently occurring DA in HH, i.e. *ack*, was only ranked 11th in HM; iii) *clarification-request*'s relative frequency (4.7%) is considerably higher in HH than in HM.

The relative frame frequency in HH dialogs is sparser than in the HM ones, meaning that the turns uttered by the machine influence the discourse topic and that the semantics of HH dialogs is more variable. The most frequent frame group comprises frames related to information exchange that is typical of the help-desk activity, including *Telling*, *Greeting*, *Contacting*, *Statement*, *Recording*, *Communication*. Another relevant group encompasses frames related to the operational state of a device, for example *Being operational*, *Change operational state*, *Operational testing*, *Being in operation*.

3. Our statistical dialog manager

We have developed a Dialog Manager (DM) based on the statistical modelization of the sequences of SLU features.A labeled corpus of dialogs is used to estimate the statistical DM. Depending on the number of these features, and thus, on the amount of information represented in them, the probability of obtaining a good model can vary. If we consider only a small number of features representing general actions in a dialog, we could obtain a well-trained model. A formal description of the proposed statistical model is as follows:

Let A_i be the output of the dialog system (the system response) at time i. Let U_i be the semantic representation of the user turn (the result of the understanding process of the user input) at time i, expressed in terms of frames. A dialog begins with a system turn that welcomes the user and offers him/her its services. We consider a dialog to be a sequence of pairs (*system-turn, user-turn*):

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

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

In this framework, we consider that, at time i, the objective of the dialog manager is to find the best system response A_i. This selection is a local process for each time i and takes into account the sequence of dialog states preceding time i. This selection is made by maximizing:

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

where set $\mathcal{A}$ contains all the possible system responses. As the number of all possible sequences of states is very large, we establish a partition in the space of sequences of states (i.e., in the history of the dialog preceding time i) by defining a data structure that we call Dialog Register and contains the information about SLU features provided by the user throughout the previous history of the dialog. Let DR_i be the dialog register at time i. Taking into account the concept of the DR, we establish a partition in the space of sequences of states such that: two different sequences of states are considered equivalent if they lead to the same DR_i. We obtain a great reduction in the number of different histories in the dialogs at the expense of a loss in the chronological information.

After applying the above considerations and establishing the equivalence relation in the histories of dialogs, the selection of the best A_i is given by:

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

Each user turn supplies the system with information about the task. However, a user turn could also provide other kinds of information, such as task-independent information. This is the case of turns in which the user provides SLU information like *Acceptance*, *Negation* and *Not-Understood*. This kind of information implies some decisions which are different from simply updating the DR_{i-1}. For that reason, for the selection of the best system response A_i, we take into account the DR that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}.

Statistical approaches must tackle the problem of modeling all the possible situations that can occur during a dialog using only the training corpus. The possibility of the user uttering an unexpected sentence must also be considered in the design of the dialog manager. We propose that, given a new user turn, the statistical dialog model makes the assignation of a system response according to the result of a classification process. The classification function can be defined in several ways. We have evaluated different definitions, providing decision trees the best results for the LUNA task.The decision tree holds a codification of the input pair (DR_{i-1}, S_{i-1}) and generates as output a decision about which of the labeled system prompts is selected for this input pair. An advantage of decision tree algorithms is that they allow automatic feature selection and their tree output provides an intuitive way to gain insight into the data.

3.1. Dialog Register representation

For the LUNA task, the DR is a sequence of 15 fields related to the specific task information. The sequence of fields is *Name*, *Organization*, *Telephone*, *Address*, *Brand-Model*, *Machine-Code* and one field for each possible problem (*Printer*, *Network_Connection*, *PC_going_slow*, *Monitor*, *Keyboard*, *Mouse*, *Virus*, *CD-DVD_player*, and *Power-Supply*).

For the DM to determine the next response, we have assumed that the exact values of the fields are not significant. They are important for access to the Database and for constructing the output sentences of the system. However, the only information necessary to determine the next action by the system is the presence or absence of values in the fields. Therefore, the information we used from the DR is a codification of this data in terms of only three values, $\{0, 1, 2\}$, for each field in the DR according to the following criteria:

- **0**: The value of the field has not been given.
- **1**: The field contains a value with a confidence score that is higher than a given threshold (a value between 0 and 1).The confidence score is given during the recognition and understanding processes.
- **2**: The field contains a value with a confidence score with a confidence score that is lower than the given threshold.

The representation defined for the input pair (DR_{i-1}, S_{i-1}) is as follows:

- The codification of the last prompt generated by the system (A_{i-1}): This information is modeled by means of a variable, which has as many bits as possible different system prompts detailed for our system (37).

$$\vec{x}_1 = (x_{1_1}, x_{1_2}, x_{1_3}, \cdots, x_{1_{25}}) \in \{0, 1\}^{37}$$

- Dialog register (DR_{i-1}): As previously stated, fifteen characteristics can be observed in the DR. Each one of these characteristics can take the values $\{0, 1, 2\}$. Therefore, every characteristic has been modeled using a variable with three bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0, 1\}^3 \; i = 2, ..., 16$$

- Task-independent information. From the information available in the labeling, we have include *Acceptance*, *Negation*, and *Not-Understood*. This information has been modeled using three variables with three bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0, 1\}^3 \; i = 17, ..., 19$$

4. Evaluation

We propose three measures to evaluate the behavior of the dialog manager developed for the LUNA task. These measures are calculated by comparing the answer automatically generated by the DM for each input in the test partition with regard to the reference answer annotated in the corpus. This way, the evaluation is carried out turn by turn. These three measures are: i) *%exact*: the percentage of answers provided by the DM that are equal to reference answer in the corresponding turn of the training corpus; ii) *%correct*: the percentage of answers provided by the DM that are coherent with the current state of the dialog although they are not the same that the reference answer; iii) *%error*: the percentage of answers provided by the DM that would cause the failure of the dialog;

The measure *%exact* is automatically calculated. On the other hand, the measures *%correct* and *%error* are manually evaluated by an expert. The expert evaluates whether the answer provided by the DM allows the correct continuation of the dialog for the current situation or whether the answer causes the failure of the dialog (e.g., the dialog manager suddenly ends the interaction with the user, a query to the database is generated without the required information, etc).

Firstly, we evaluated the behavior of a DM learned using only the 200 dialogs from the HM corpus. A 5-fold cross-validation process was used to carry out the evaluation of this manager. The corpus was randomly split into five subsets of 472 samples (20% of the corpus). Our experiment consisted of five trials. Each trial used a different subset taken from the five subsets as the test set, and the remaining 80% of the corpus was used as the training set. A validation subset (20%) was extracted from each training set. The Weka machine learning software was chosen to learn C4.5 decision trees (called *J48* in Weka). The number of different possible system answers in the corpus was 36. The average of different (DR, S) pairs in the training sets was 431. Table 1 shows the results of the evaluation of the HM Dialog Manager. The codification developed to represent the state of the dialog and the good operation of the decision tree make it possible for the answer generated by the manager to agree with the reference answer by a percentage of 97.05%. Finally, the number of answers generated by the DM that can cause the failure of the system represents only a 1.43% percentage. An answer that is coherent with the current state of the dialog is generated in 98.58% of cases. These last two results also demonstrate the correct operation of the methodology.

Secondly, we repeated the same experiment for the case of the HH corpus. This corpus was randomly split into five subsets of 1,549 samples (20% of the corpus) and also a 5-fold cross-validation process was used for its evaluation. The number of different possible system answers in the corpus was 37. The average of different (DR, S) pairs in the training sets was 1,477. Table 1 shows the results of the evaluation of the HH Dialog Manager. As can be observed, the developed DM also adapts to the specific characteristics of the HH dialogs, achieving a 92.33% value for the *%correct* measure. This result has to be emphasized due to the complexity of the HH dialogs, as stated in Section 2.2. The main factor for the decrement of the *%error* measure (compared to the HM Manager), was the confusion of the *Out of the task* topics with the system prompts that are related to the task.

Finally, we learned a DM with the total 200 HM dialogs and evaluated the resulting DM using the total 200 HH dialogs. This experimentation was designed to evaluate if a model learned with HM dialogs can detect the structure of spontaneous real user conversations. The main challenge of this experiment is that the *Out of the task* prompts make up the 27.31% of the HH corpus and are not present in the HM corpus. Therefore, only a maximum of 72.69% can be achieved for the *%exact* value. Table 1 shows the results of the evaluation of the HM-HH Dialog Manager. We can see that the DM provides a 75.11% value of coherent answers for the system turns in the HH corpus. The main errors are caused by the re-asking of information due to the changes in the order of the different parts in both kinds of dialogs, and the premature ending of the dialog mainly due to the presence of the *Out of the Task* turns.

	%exact	%correct	%error
HM Dialog Manager	97.05%	98.57%	1.43%
HH Dialog Manager	84.26%	92.33%	7.67%
HM-HH Dialog Manager	61.45%	75.11%	24.89%

Table 1: Results of the evaluation of the DMs developed for the different experimentations

5. Conclusions

In this paper, we have presented a corpus-based methodology for the development of statistical dialog managers. We have developed a detailed representation of the user SLU features, that allows the system to automatically generate a specialized answer that takes into account the current situation of the dialog. From this representation, a classification methodology based on decision trees is used in order to generate the system responses. We have described an evaluation of this methodology within the framework of the European LUNA project. One of the main conclusions of this study is that our statistical dialog methodology not only adapts separately to the very different nature of the human-machine and human-human dialogs acquired for this project, but also it is possible to successfully detect the task-related information that is present in the human-human dialogs by learning a dialog manager using only human-machine dialogs. As a future work, an evaluation of the behavior of the system using real users is going to be made to compare the results with those presented in this paper. Additional task independent information is also going to be incorporated to the model to try to detect between the different out of the task turns that are present in the HH corpus.

6. Acknowledgements

Work partially funded by the European commission LUNA project contract 33549 and the Spanish MEC and FEDER under contract TIN2008-06856-C05-02.

7. References

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

[2] T. Paek and R. Pieraccini, "Automating spoken dialogue management design using machine learning: An industry perspective," *Speech Communication*, vol. 50, no. 8-9, pp. 716–729, 2008.

[3] F. Torres, E. Segarra, and E. Sanchis, "User simulation in a stochastic dialog system," *Computer Speech & Language*, vol. 22, no. 3, pp. 230–255, 2008.

[4] V. Rieser and O. Lemon, "Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz data: Bootstrapping and Evaluation," in *Proc. of ACL-08: HLT*, Columbus, USA, 2008, pp. 638–646.

[5] J. Williams and S. Young, "Partially Observable Markov Decision Processes for Spoken Dialog Systems," *Computer Speech and Language*, vol. 21(2), pp. 393–422, 2007.

[6] D. Griol, L. F. Hurtado, E. Segarra, and E. Sanchis, "A statistical approach to spoken dialog systems design and evaluation," *Speech Communication*, vol. 50, no. 8-9, pp. 666–682, 2008.

[7] M. Dinarelli, S. Tonelli, A. Moschitti, and G.Riccardi, "Annotating Spoken Dialogs: from Speech Segments to Dialog Acts and Frame Semantics," in *Proc. of EACL 2009 Workshop on Semantic Representation of Spoken Language*, Athens, Greece, 2009, pp. 34–41.