A conversational agent for information retrieval based on a study of human dialogues
Alain Loisel, Guillaume Dubuisson Duplessis, Nathalie Chaignaud, Jean-Philippe Kotowicz, Alexandre Pauchet

To cite this version:

HAL Id: hal-00959088
https://hal.archives-ouvertes.fr/hal-00959088
Submitted on 14 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A CONVERSATIONAL AGENT FOR INFORMATION RETRIEVAL
BASED ON A STUDY OF HUMAN DIALOGUES

A. Loisel, G. Dubuisson Duplessis, N. Chaignaud, J-Ph. Kotowicz and A. Pauchet
LITIS - EA 4108, INSA Rouen, Avenue de l’Université - BP 8,76801 Saint-Étienne-du-Rouvray Cedex, France
{name}@insa-rouen.fr

Keywords: Conversational Agent, Dialogue System, Mixed-Initiative Interaction

Abstract: This study strives to improve medical information search in the CISMÆF system by including a conversational agent to interact with the user in natural language. Experimentation has been set up to obtain human dialogues between a user dealing with medical information search and a CISMÆF expert refining the request. We extend the GoDiS dialogue system with dialogue strategies in order to support system digressions. A model of an artificial agent has been implemented.

1 INTRODUCTION

The CISMÆF system (www.cismef.org (Darmoni et al., 2000)) aims to provide access to resources in the medical domain for healthcare professionals, medical students but also patients. CISMÆF proposes a graphical user interface and a query language to build queries using a controlled vocabulary called MeSH (Medical Subject Headings).

However, users face problems with its use: (i) The query language is complex and users have to know the CISMÆF terminology; (ii) Users are confused by the presentation of query results.

Natural language dialogue interfaces provide a good solution to these problems. It has been argued that a mixed-initiative conversational agent provides an easier, interactive and natural access to information (Androutsopoulos et al., 1995).

Our assumption is that the analysis of H-H interactions brings essential hints to design a H-C dialogue system even though the nature of these interactions is not strictly identical. We have set up an experiment to obtain a corpus of human dialogues between a user playing the role of a patient who searches medical information and a CISMÆF expert trying to help him to build the query and to obtain answers. The hand-analysis of these interactions highlights their discursive structure and their linguistic features. Our dialogue model is based on this analysis of the corpus. We have shown that the concepts of GoDiS (Larson, 2002) partly meet our needs. We present in this article the essential elements of the corpus analysis and the arguments that led us to adopt this system.

2 APPROACHES TO DIALOGUE SYSTEM DESIGN

2.1 Approaches to Dialogue Management

Several approaches to dialogue management exist and no approach clearly dominates the others. The simplest one is the finite-state (FS) approach (McTear, 2004) that represents the structure of the dialogue as a finite-state transition network. In practice, this approach turns out to be rigid and limited to system-directed dialogue. The frame-based approach considers the dialogue as a process of filling in a frame which contains a series of slots (Aust et al., 1995). It is less rigid than the FS approach. However, the possible contributions of the system are fixed in advance. The plan-based approach (Allen and Perrault, 1980) comes from classical AI. It combines planning techniques such as plan recognition with ideas from speech act theory. This approach is rather complex from a computational perspective and requires advanced Natural Language Understanding (NLU) components in order to infer the intentions of the speaker. The logic-based approach represents the dialogue and its context in some logical formalism and takes advantage of mechanisms such as inference (Hulstijn, 2000). Most works are still on a theoretical level with this approach. More recent approaches aim to automatically learn dialogue policies with machine learning techniques such as reinforcement learning (Frampton and Lemon, 2009). These
approaches require an extensive effort of annotation since a large amount of annotated data is necessary.

2.2 GoDiS/IBiS Systems

The Question Under Discussion (QUD) model (Ginzburg, 1996) is a theory of dialogue which aims to precisely analyze the properties of issues. Ginzburg comprehends dialogues as moves (e.g., ask) that modify the conversational board.

Based on the QUD theory, GoDiS and the IBiS family of systems (Larsson, 2002) are computational dialogue models which deal with inquiry-oriented dialogue. They use a simplified version of the semantics of the questions. Three types of question are possible: yes-no questions (\(?P\)), wh-questions (\(?x.P(x)\)) and alternative questions (\(\{?P_1, ?P_2, . . . , ?P_n\}\)). IBiS also considers taskAction which is task-dependent. Question and taskAction are collectively designated as issue. Three dialogue moves are available, namely ask (to ask a question), answer (to answer a question) and request (to request an action from the system).

Issues are integrated in dialogue plans which are sequences of abstract actions. GoDiS is Information-State-Update-based (Larsson and Traum, 2000; Bos et al., 2003) and then uses an information state (IS), close to the conversational board. This complex structure comprises a private part and a shared part. The private part corresponds to the inner state of the agent (one interlocutor) and the shared part defines the conversational board, which memorizes shared information between the two interlocutors. The update rules and the selection rules can change the IS. According to the initial dialogue move, the reactive dialogue moves are generated.

The main benefit of GoDiS is the notion of accommodation (Larsson, 2002) which brings flexibility to the dialogue. This mechanism enables the user not to answer the current issue, to correct a previous answer, to answer an unasked question or to move to another dialogue plan. However, GoDiS and IBiS only represent a simple task with a limited number of dialogue moves. We discuss in more details the advantages and drawbacks of GoDiS in section 3.3.

3 CORPUS COLLECTION AND ANALYSIS

3.1 Corpus Collection

An experiment has been set up to obtain human dialogues between experts and users, dealing with medical information search. The users were voluntary members of the LITIS laboratory (secretary, PhD students, researchers and teachers), who wanted to obtain answers about medical inquiries. These users represent the targeted audience for the system since they are not medical specialists. Experts were two members of our project, trained in the CISMEx system and its terminology. The experimentation took place as follows: one expert and one user were facing a computer using the advanced search interface and recording all the queries along with their answers in a log. The expert was in charge of conducting the search by conversing with the user and verbalizing each action, inquiry and answer. The experimentation ended when relevant documents were found or when it seemed that no answer existed in the system. Having two distinct experimenters increased the variability of the recorded dialogues. A textual corpus has been constructed from the transcription of the 21 dialogues between the two experts and the 21 volunteers. It contains 37 000 words.

3.2 Corpus Analysis

We have hand-analyzed the corpus at 3 levels (Loisel et al., 2008): (i) An analysis of dialogue acts (as contextualized speech acts (Bunt, 1996)): a list is proposed according to the illocutionary goals of the utterances of the corpus. (ii) An analysis of sub-dialogues which considers the dialogue as a sequence of stages. (iii) An analysis of issues: we listed issues appearing in each sub-dialogue.

Dialogue acts (Bunt, 1996) firstly aim to modify the context and are interpreted according to the current situation. This notion is adequate to our study. Each utterance of our corpus can be broken down into segments associated with a dialogue act. A list of dialogue acts has been built according to linguistic features found in the corpus. This taxonomy is adapted from (Weisser, 2003) who synthesized existing annotating schemes. Our taxonomy is close to existing ones such as DAMSL (Core and Allen, 1997).

We set up a list of 36 dialogue acts containing classic ones (e.g., Answer, Inform, RequestInfo) and grounding acts (Traum, 1994) (e.g., Accept, Acknowledge, Confirm, Refuse).

We discovered and extracted a global structure for the corpus dialogues. This structure consists of sub-dialogues combined with sequence relations. We established a list of 14 sub-dialogues like opening, request formulation and evaluation of the results sub-dialogues. Their main characteristic is that they are opportunistic and some of them are optional. For instance, the opening sub-dialogue which includes the greetings between the user and the system may be
skipped. The study of the dialogues also shows digressions from the expert or the user with incidental dialogues. For example, a definition sub-dialogue may intervene opportunistically to bring a definition of a CISMeF term to the user.

We established a list of issues for each sub-dialogue. We simplify the problem by considering that issues are only a means to obtain CISMeF terms. That is why we restrict our description semantics to the first-order logic. A total of 44 issues have been listed in the analysis.

3.3 Discussion on the Corpus Analysis

The analyses of the corpus showed that we were facing the same kind of dialogues than IB1S, i.e., inquiry-oriented dialogue. Then, our idea is to reuse this already existing system with some adaptations.

The IB1S approach is well adapted to the dialogues of our corpus. Firstly, it is based on an explicit task which does not require much reasoning on the users’ intentions. The sub-dialogues extracted from the corpus combined with the corresponding issues match dialogue plan in IB1S. Secondly, we reckon that the sub-dialogues could be left explicitly or implicitly. This feature is caught by the accommodation mechanism. Finally, IB1S dialogue act list is close to our taxonomy. In particular, our set of grounding acts corresponds to the Interactive Communication Management (ICM) (Larsson, 2002) in IB1S.

However, these acts are not sufficient. The dialogue acts Inform, Offer and Suggest are missing. They allow the expert to propose relevant information opportunistically according to the current results.

Strategies of dialogue manage turns between the user and the system to efficiently conduct the dialogue (Caelen, 2003). Caelen distinguishes five strategies of dialogue: (i) The directive strategy is used to lead entirely the user. (ii) The reactive strategy is used when the system executes the orders of the user. (iii) The constructive strategy introduces digressions to temporarily leave the current goal for a new one. This strategy is adopted to present examples or a previous experience useful to the current situation. (iv) The cooperative strategy adjusts the current goal to the user’s one. The system tries to fit the goal of the user by suggesting new information opportunistically or offering choices, while staying in the same topic. (v) Finally, the argumentative strategy appears when the user disagrees with the goal proposed by the system. This strategy is mostly found in argumentative dialogues but our application is not concerned with it.

If we take a closer look at IB1S, we realize that the directive and reactive strategies of dialogue are available with the set of dialogue moves provided. As a matter of fact, the directive strategy consists in the classic progress in the dialogue plan. Besides, reactive strategies are managed. Indeed, user-initiated digressions are allowed with ask-move and request-move. Moreover, the accommodation mechanism enables this kind of digressions with answer-move.

Cooperative and constructive strategies can be collectively viewed as system-initiated digressions. This kind of digressions is not possible with the current IB1S system since dialogue plans are rigid and predetermined. However, it can be interesting to use system digressions in an information search dialogue. For instance, we can imagine a system that proactively presents information related to the user request. That is why one of our goals is to integrate cooperative and constructive strategies into IB1S.

4 THE COGN1-CISMeF AGENT

4.1 Integration of Cooperative and Constructive Strategies in IB1S

The dialogue manager models the sub-dialogues observed in the corpus with a plan library, represents the common ground and controls the information state (IS). While it is based on IB1S, it additionally includes a model of questions with several satisfactory answers, question accommodation and action accommodation, intentional relations and finally dialogical strategies. In this section, only parts of the plan library and dialogical strategies are described.

4.1.1 The Plan Library

Our model uses two kinds of plans described in the first-order logic using "?" to represent the questions: (i) question plans (PlanQ) which aim at answering inquiries by returning data; (ii) action plans (PlanA) which perform sequences of actions. These plans use a list of actions coming from IB1S: Findout(q) allows the system to ask a question q by generating the ask-move. The system asks this question iteratively until it is answered or cancelled. AssumeAction(a) adds an action into the IS. Forget and ForgetAll delete knowledge from the IS. This list is not exhaustive.

For instance, figure 1 corresponds to the action plan DocumentSearch which begins with the three first steps of a search: the query formulation, the query building and the display of the current query. Then, a yes-no question is asked by a Findout so that the user can validate the query. If the response is negative, the system forgets the previous results and re-
PlanA(DocumentSearch, 
   (IfThen(not AddKeyword(m)))
   (AssumeAction(QueryFormulation), 
   AssumeAction(QueryBuilding), 
   AssumeAction(QueryDisplay), 
   FindCooperativePlanAction(Q,Ps,PlanAction), 
   NotEnoughDocuments(TooManyDocuments), 
   TooManyDocuments(), 
   ExecuteCooperativeAction(Q,Ps,PlanAction) 
   AddPlan(PlanAction)) 
   )

Figure 1: Action plan DocumentSearch

initiates the plan with the action QueryBuilding to refine the query. Otherwise, the query is performed and the results are appraised. Unless there is an explicit exit dialogue act uttered by the user or by the system, or an exit planned by the current plan, the system remains in this plan. All these actions are linked by satisfaction-precedence relations.

4.1.2 The Dialogical Strategies

The Cooperative strategies modify plans in order to propose suggestions or help. The dialogue has to remain in the same context. The current plan is always active and the system proposes to the user new information acquired dynamically during the search. Practically speaking, it dynamically builds new issues (questions, actions or propositions) and adds them to the current plan in shared/plan, while keeping the other fields of the IS unchanged.

A new action called CooperativeAction is created in three ways: (i) CooperativeAction(?) to ask a new question to the user; (ii) CooperativeAction(action) to add a new taskAction; (iii) CooperativeAction(p(x)) to propose an answer or a suggestion to a pending question. The system can also answer its own questions by consulting the database. This predicate can be added at any place into a plan, where it is relevant to propose suggestions to the user.

When this action is found in the current plan, an update rule called ExecCooperativeAction is applied (Figure 2). It runs the procedure FindCooperativePlanAction which finds new information to be presented to the user. Finally, the rule adds a new plan action to the top of the stack private/plan.

Figure 2: Update rule ExecCooperativeAction. PRE stands for preconditions on the IS. EFF stands for effects on the IS.

PlanA(ResultEvaluation, 
   (IfThen(n isTooLow)
   (Inform(NotEnoughDocuments), 
   CooperativeAction(7x.Issue(x) NotEnoughDocuments)), 
   ForgetIssue(), 
   ForgetAction(), 
   AssumeAction(DocumentSearch), 
   Assume(CorrectNbDocuments), 
   Confirm(ResultEvaluation)) 
   )

Figure 3: Action plan ResultEvaluation

For example, one major issue for a search engine is to find the correct number of documents to be presented to the user. This issue concerns the ResultEvaluation (figure 3) plan which labels the number of documents found with TooManyDocuments or NotEnoughDocuments. This label is added to the IS through the action plan Inform. If this number is too high, the plan calls the CooperativeAction plan, to refine the query by proposing suggestions dynamically pushed in private/bel and private/plan. A similar plan is executed when no or very few documents are found. Then, the search has to be launched again, taking into account the user’s answers to these suggestions. By plan accommodation, the user can add new terms or address new issues to the query. The plans ForgetAction and ForgetIssue are used to clean the IS. The plan AssumeAction is performed to relaunch the new query. Finally, if the number of documents is correct, the result can be refined by sorting documents to propose the most relevant first.

The Constructive strategies aim to bring explanation sequences during the dialogue. It diverts the system from its current plan: new information has been acquired that makes the system propose digressions, hints or examples. This new information come from
4.2 The Task Model

The task model gives access to advanced functionalities of CISMef to the other modules composing COGNI-CISMef. It includes: (i) The CISMef terminology which enables the system to determine subheadings, definitions, hypernyms and hyponyms associated with a term. It also contains a list of terms used by patients in order to detect medical keywords in user utterances. (ii) A query builder which uses terms detected by the NLU component. It builds a query as a human expert would do in the graphical interface. (iii) A result interpreter which refines the query if necessary. It includes various search strategies (e.g., if the query returns too many documents, it is necessary to specify it by adding new terms). This interpreter suggests to the dialogue manager to add relevant issues in order to restrict the query. (iv) The CISMef engine which runs the query so as to access to medical documents.

4.3 Natural Language Understanding and Generation

COGNI-CISMef receives the user’s utterance in text-written natural language in French and carries out two main analyses: a semantic one and a pragmatic one.

The semantic analysis starts with a recognition of textual units. Afterwards, recognition and labeling of CISMef terms is carried out. These terms are identified by the database via the task model.

The pragmatic analysis is driven by a rule-based dialogue act interpreter. The rules used are based on linguistic markers, such as the tense and the mode of the verb, interrogation markers, acceptance markers, etc., associated with the immediate dialogue co-text (the previous statement of the dialogue). Finally, the results of these analyses are concatenated to form a proposition usable by the dialogue manager.

The sentence generator is rather simple since it translates a logical proposition coming from the dialogue manager into an instantiated incomplete sentence. However, the system uses mechanisms that produce more natural utterances. For example, a random generator diversifies the most frequent utterances as greetings, agreements, etc.

5 DIALOGUE EXAMPLE

It is possible to perform a full dialogue by selecting appropriate terms. Listing 1 presents a real commented dialogue between the COGNI-CISMef system (CC) and a user (U).

Listing 1: Example of dialogue with COGNI-CISMef.

\begin{verbatim}
CC: Welcome to Cognicismef. What can I do for you? % Opening sub-dialogue + choice sub-dialogue
U: Give me a definition of paludism % Definition sub-dialogue
CC: The term means: "Also called malaria [...]" Now, what can I do for you? % Answer + come back to the choice sub-dialogue
U: Search documents about it % Search documents about it
CC: OK. Do you want to add the keyword "paludism"? % Query formulation sub-dialogue
CC: OK. Do you want to add something else? % Issue establishment + query building sub-dialogue (dependance relation)
U: Yes
CC: OK. Do you want to add something else? % Are there therapeutics?
CC: OK. Do you want to add the meta-term "therapeutic" or the subheading "therapeutic"? % Query building sub-dialogue
U: What is a subheading? % Definition sub-dialogue
CC: Subheadings are general concepts associated with a keyword to precise its meaning
\end{verbatim}
6 CONCLUSION

This article presents the design of a human-computer dialogue system, based on QUD theory and IBIS, from corpus collection to implementation. Our conversational agent requires enhancements: at least, the semantic analysis needs a bigger lexicon. Then, the system will be able to be evaluated. It will consist in a comparison of the users’ requests using CISMEOF with those proposed by the librarian of the medical library of the hospital of Rouen and those built by users using COGNI-CISMEOF. We will measure the improvements (in terms of precision and recall) respectively of the COGNI-CISMEOF system and the librarian compared to the request using only CISMEOF.

REFERENCES


