6th IJCAI Workshop on

Knowledge and Reasoning in Practical Dialogue Systems

Pasadena, California
July 12, 2009

Proceedings edited by Arne Jönsson, Jan Alexandersson, David Traum and Ingrid Zukerman

6th IJCAI Workshop on KNOWLEDGE AND REASONING IN PRACTICAL DIALOGUE SYSTEMS

This is the sixth workshop on Knowledge and Reasoning in Practical Dialogue Systems, held at IJCAI-2009 in Pasadena.[1] Each workshop in this series has focused on a different aspect of dialogue systems. The first workshop, which was organised at IJCAI-99 in Stockholm,[2] was aimed at studying the need for knowledge and reasoning in dialogue systems from theoretical and practical perspectives. In addition to innovative aspects of research, emphasis was placed on the importance of implemented dialogue systems as test-beds for evaluating the usefulness of theories and ideas, and on improvements in the ability of practical systems to support natural and efficient interactions. The second workshop took place at IJCAI-2001 in Seattle.[3] It focused on multimodal interfaces: the coordination and integration of multimodal inputs and the ways in which multimodal inputs reinforce and complement each other, and the role of dialogue in multimodal interaction. The focus of the third workshop was the role and use of ontologies for developing flexible, adaptive, user-friendly and enjoyable multi-modal dialogue systems. This workshop was held at IJCAI-2003 in Acapulco.[4] The fourth workshop, held at IJCAI-2005 in Edinburgh,[5] emphasized adaptivity in dialogue systems, including research in dialogue management, adaptive discourse planning and automatic learning of dialogue policies. The fifth workshop was held at IJCAI-2007 in Hyderabad, India, and focused on dialogue systems for robots and virtual humans.[6]

The current workshop has a focus on the challenges posed by novel applications of practical dialogue systems. It includes presentations and discussion of research on novel applications of dialogue systems, probabilistic reasoning and resource integration, dialogue frameworks, and evaluation and empirical methods.

These workshop notes contain 6 long papers and 7 short papers that address these issues from various view-points. The papers offer stimulating ideas, and we believe that they form the basis for fruitful discussions during the workshop, and further research in the future.

The program committee consisted of the colleagues listed below. Without the time spent reviewing the submissions and the thoughtful comments provided by these colleagues, the decision process would have been much more difficult. We would like to express our warmest thanks to them all. Additional thanks to Thomas Kleinbauer and Andreas Eisele for help with LaTeX issues.

[1] http://www.ida.liu.se/~arnjo/Ijcai09ws/

[2] http://www.ida.liu.se/~nlplab/ijcai-ws-01/. Selected contributions have been published in a special issue of ETAI, the Electronic Transaction of Artificial Intelligence http://www.ida.liu.se/ext/etai/

[3] http://www.ida.liu.se/~nlplab/ijcai-ws-01/

[4] http://www.ida.liu.se/~nlplab/ijcai-ws-03/

[5] http://www.csse.monash.edu.au/~ingrid/IJCAI05dialogueCFP.html

[6] http://people.ict.usc.edu/~traum/ijcai07ws/

Program Committee

Dan Bohus, Microsoft Research, USA
Johan Bos, Università di Roma "La Sapienza", Italy
Sandra Carberry, University of Delaware, USA
Kallirroi Georgila, Institute for Creative Technologies, USA
Genevieve Gorrell, University of Sheffield, UK
Joakim Gustafson, KTH, Sweden
Yasuhiro Katagiri, NIH, USA
Kazunori Komatani, Kyoto University, Japan
Staffan Larsson, Götteborg University, Sweden
Anton Nijholt, University of Twente, The Netherlands
Tim Paek, Microsoft Research, USA
Antoine Raux, Honda Research Institute, USA
Candace Sidner, Mitsubishi Electric Research Labs, USA
Amanda Stent, Stony Brook University, USA
Marilyn Walker, University of Sheffield, UK
Jason Williams, AT&T, USA

Organizing Committee

Arne Jönsson (Chair)
Department of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden
email: arnjo@ida.liu.se

Jan Alexandersson (Co-chair)
German Research Center for Artificial Intelligence, DFKI GmbH
Stuhlsatzenhausweg 3
66 123 Saarbrucken, Germany
email: janal@dfki.de

David Traum (Co-chair)
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292 USA
email: traum@ict.usc.edu

Ingrid Zukerman (Co-chair)
Faculty of Information Technology
Monash University
Clayton, Victoria 3800, Australia
email: Ingrid.Zukerman@infotech.monash.edu.au

Table of Contents

Workshop Program

8:30 Welcome, introductions and opening discussion

8:45 **Session 1: Novel applications**
8:45 Bullying and Debt: Developing Novel Applications of Dialogue Systems
Keeley Crockett, Zuhair Bandar, James O'Shea, David Mclean
9:15 An Integrated Authoring Tool for Tactical Questioning Dialogue Systems
Sudeep Gandhe, Nicolle Whitman, David Traum, Ron Artstein
9:45 Discussion

10:00 Coffee break

10:30 **Session 2: Probabilistic reasoning and resource integration**
10:30 Interpreting Two-Utterance Requests in a Spoken Dialogue System
Ingrid Zukerman, Enes Makalic, Michael Niemann
11:00 Integrating Spoken Dialog with Bayesian Intent Recognition: A Case Study
Ronnie W. Smith, Brian Adams, Jon C. Rogers
11:15 Open-World Dialog: Challenges, Directions, and Prototype
Dan Bohus, Eric Horvitz
11:45 A 3D Gesture Recognition System for Multimodal Dialog Systems
Robert Neßelrath, Jan Alexandersson
12:00 Discussion

12:15 Lunch

1:45 **Session 3: Dialogue frameworks**
1:45 Dialog Modeling Within Intelligent Agent Modeling
Marjorie McShane, Sergei Nirenburg
2:15 The Companions: Hybrid-World Approach
Alexiei Dingli, Yorick Wilks, Roberta Catizone, Weiwei Cheng
2:30 A Set of Collaborative Semantics for an Abstract Dialogue Framework
Julieta Marcos, Marcelo A. Falappa, Guillermo R. Simari
2:45 Discussion

3:00 Coffee break

3:30 **Session 4: Evaluation and empirical methods**
3:30 Automatic handling of Frequently Asked Questions using Latent Semantic Analysis
Patrik Larsson, Arne Jönsson
4:00 Seeing What You Said: How Wizards Use Voice Search Results
Rebecca J. Passonneau, Susan L. Epstein, Joshua B. Gordon, Tiziana Ligorio
4:15 Subjective, But Not Worthless – Non-linguistic Features of Chatterbot Evaluations
Pawel Dybala, Michal Ptaszynski, Rafal Rzepka, Kenji Araki
4:30 SpeechEval – Evaluating Spoken Dialog Systems by User Simulation
Tatjana Scheffler, Roland Roller, Norbert Reithinger
4:45 Discussion
5:00 Closing discussion

Bullying and Debt: Developing Novel Applications of Dialogue Systems

Keeley Crockett, Zuhair Bandar, James O’Shea, David Mclean

The Intelligent Systems Group, Department of Computing and Mathematics, Manchester Metropolitan University, Chester Street, Manchester, M1 5GD, UK. email K.Crockett@mmu.ac.uk).

Abstract

Human Resource (HR) departments in organisations provide human advisors to give guidance on the policies and procedures within the organisation. Policies and procedures are often complex and written using a particular legal vocabulary which is sometimes difficult for employees to interpret and understand in relation to their personal circumstances. The fairly static nature and high support costs of such policies means that they lend themselves to automation. Conversational Agents are dialogue systems that have the ability to converse with humans through the use of natural language dialogue in order to achieve a specific task. Traditionally, they utilise pattern matching algorithms to capture specific attributes and their values through dialogue interaction with a user. This is achieved through the use of scripts which contain sets of rules about the domain and a knowledge base to guide the conversation towards achieving the task. Such systems are ideal for providing clear and consistent advice 24 hours a day and allow the employees to be able to ask questions in natural language about the policies/procedures which are relevant to their personal situation. This paper presents an overview of a methodology for constructing text based conversational agent advisors. Two case studies which employ this methodology are introduced and evaluated. Finally research issues in designing text based conversational agents as human advisors are addressed and areas of current research are discussed.

Introduction

In organisations human advisors are available to provide guidance to employees on the organisation’s policies and procedures. For example, if a member of staff wanted to find out their holiday entitlement they would contact their Human Resources (HR) Department or line manager. Not all policies are easy to discuss with a human being and the employee may feel sensitive about asking for advice in certain areas such as advice on maternity rights, failings during the probation period, or what to do if they feel they are being bullied or harassed. Whereas the policies and procedures may themselves be readily available as hard or soft copy documents, they are lengthy, complex and written using a particular legal vocabulary. Due to this complexity, an individual may not be able to apply the policy or procedure to their personal situation and any problems they have continue to develop further. The fairly static nature and high support costs of such policies means that they lend themselves to automation. Many of these problems can be resolved using conversational agents.

A conversational agent (CA) is an agent which can fully participate in natural language dialogue (Massaro et al, 2000). The CA is a type of dialogue system and within this paper we use the term interchangeably. Ideally, the CA exhibits certain aspects of intelligent behaviour such as the ability to perceive the environment around it and have knowledge about the current state of this environment. The CA will also have the ability to reason and pursue a course of action based on its own current position in the environment and its interactions with humans and other agents. An automated and interactive conversational agent system could provide anonymous and 24-hour access to these policies/procedures and allow the employees to be able to ask questions in natural language about the policies/procedures which are relevant to their personal situation. At the same time, the advice given by the conversational agent would always be consistent, appropriate and valid whilst the agent can be designed to exhibit sympathetic or compassionate behaviour to a particular circumstance. A further strength is that they can be tailored to behave in a way that reflects an organization’s culture and to have distinctive personalities.

This paper first introduces a brief history of conversational agents and then goes on to propose a methodology for constructing conversational agents. Two novel applications of conversational agents which act as advisors on the subject of bullying and harassment and student debt problems are then described. Finally some of the main issues in developing conversational agents are highlighted by examining current research in both the scripting and evaluation of such agents.

Conversational Agents

The idea that a computer could actually engage in a conversation with a human being was thought to be the subject of science fiction for many years. In 1950 British mathematician and code-breaker Alan Turing published a seminal paper, Computing Machinery and Intelligence which discussed the question “Can machines think?”

(Turing, 1950). Since then the ability to create a computer that could communicate using natural language has been one of the main challenges of computer scientists worldwide. This has led to the development of conversational agents, computer based agents. The implication of this technology, even whilst still in its infancy is that a machine rather than a human operator can engage in a conversation with a person to try and solve their problem(s). The best known early conversational agent was Eliza (Weizenbaum, 1966). Modelled on Rogerian Therapy, Eliza used questions to draw a conversation out of the user. However the main criticism of ELIZA was the program's lack of an internal world model that could influence and track conversation (Mauldin, 1994). An advancement made on ELIZA was known as PARRY (Colby, 1975), an agent with a personality that could admit ignorance, be controversial and even be humorous. PARRY simulated paranoid behaviour by tracking its own internal emotional state throughout the conversation. Colby (Colby, 1975), subjected PARRY to blind tests with doctors questioning both the program and three human patients diagnosed as paranoid. Reviews of the transcripts by both psychiatrists and computer scientists showed that neither group did better than chance in distinguishing the computer from human patients. A.L.I.C.E. (Alice, 2009) uses pattern recognition combined with symbolic reduction to understand more complex forms of input and draws on a large knowledge base to formulate an appropriate response. The A.L.I.C.E. AI Foundation promotes the use of Artificial Intelligence Mark-up Language (AIML) for creating a knowledge base organised into categories and topics.

Another more recent conversational agent is known as Infochat (Michie, and Sammut, 2001). The agent essentially implemented an interpreter for a language known as Pattern Script which was designed for developing and testing conversational agents. The agent works by using a suite of scripts to allow the system to mimic some form of behaviour. Each script is written for a specific context and composed of a number of rules which can fire or retain information about the conversation in memory. A number of complex parameters are used to ensure that the correct rule fires. Writing scripts for Infochat is a craft in itself and requires a sound knowledge of the scripting algorithm and good understanding of the English language. In order to get specific rules to fire, every combination of the words that the user might utter in response must appear within the pattern part of the rules. Wildcards are used to alleviate the problem of having to write down all possibilities that a human may respond with (which is an impossible task). Hence, the scripting process is very labour intensive, however as long as the scripting methodology is followed; the results are impressive compared with other conversational agents (Bickmore and Giorgino, 2006).

A significant proportion of conversational agents research has been dedicated towards embodied agents where the features and characterises of a visual agent are taken as much as the actual conversation itself in determining the users ability to converse in a human like manner (Cassell et al, 2000). An embodied agent can be defined as an agent that performs tasks, such as question answering, through a natural-language style dialogue. The interface is usually a human face, which is capable of facial expressions and gestures in addition to words in order to convey information. The purpose is to make a computer application appear more human like and become more engaging with the user. While substantial amounts of work (Iacobelli and Cassell, 2007; Cassell, 2007; Xiao, 2006; Massaro, 2004) have gone into evaluating such interfaces in terms of features such as visual appearance, expressiveness, personality, presence, role and initiative, less attention has been applied to the evaluation of the actual conversation.

Conversational agents are also being used in other areas where the aim is to move away from complete human dependence for help and advice. One such area is the use of conversational agents as natural language interfaces to relational databases (Popescu et al, 2003; Owda et al, 2007; Puder et al, 2007). Owda et al (Owda et al, 2007) proposes the use of goal oriented conversation to provide the natural language interface and helps disambiguate the user’s queries, by utilizing dialogue interaction. The system also incorporates knowledge trees which are used to direct the conversational agent towards the goal (i.e. query generation from natural language). The system provides naive users with an easy and friendly interface to relational databases instead of the need to be familiar with SQL to generate both simple and complex queries.

Whilst research into mechanisms for scripting and developing CAs continues, a number of commercial CA’s are available. VirtuOz, use conversational agents for customer support – such as in detecting business leads, online sales, advice and recommendations, customer support and online helpdesks and helping with customer loyalty programmes (VirtuOz 2009). Their conversational agents are each given a specific ‘mission’ such as technical sales or assistance, and their knowledge base contains domain-specific information which they draw on during textual conversations, online, via email or SMS. An example is an agent called Lea created for Voyages-SNCF.com, who responds to questions about train services, and has reportedly halved the number of customer service emails (VirtuOz 2009). Victauri LLC has developed Stauri, a number of software tools which allow users to develop ‘knowledgebots’ and virtual characters for websites (VirtuOz 2009). The various tools allow development for particular purposes, such as for answering questions, finding relevant information, service, training and educational agents, and CAs which can summarise documentation and answer questions on the content. Convagent Ltd (Convagent, 2009) in conjubction with members of the Intelligent Ssytems Group at ?Manchester Metropolitan University, developed Adam, a student debt advisor who can capture information from students about why they are in debt, and guide them through options for paying outstanding university fees (Convagent, 2009).

Adam is designed to supplement the telephone helpline, which is available during working hours only, and may be oversubscribed at certain times of the year in large universities. The nature of support offered has required Adam to cope with upset and angry students, and deal with abusive language. A more comprehensive description of the technology and an evaluation of 'Adam' will be discussed later in this paper.

A Methodology for Constructing Conversational Agents

The conversational agents used within this work employ a novel pattern matching methodology to identify key words in user input. Once key words have been identified, the conversational agent will try to match these to a predetermined pattern, and issue the corresponding response to the user. This can be a continuation of the dialogue (i.e. to get clarification, ask for more information etc.) or it can be the result of a process triggered by the conversational agent. The methodology organises sets of keywords and patterns into different contexts, which either guide the user through different contexts using dialogue, or have a method for identifying when a user wants to change context. Focusing on one context at a time narrows down the range of language which the user will input, and makes it easier for the agent to determine what the user requires.

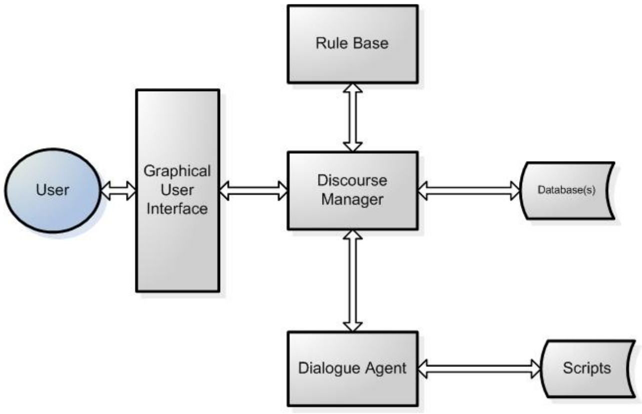


Figure 1 displays the main components of the CA architecture.

Fig. 1 .Generic architecture

Each component will now be briefly described:

- The *Discourse Manager* is the central component of the CA, providing communication with a rule base, dynamic graphical user interface (GUI) and the conversational agent itself (CA). When events take place in the GUI, the controller requests a response from the rule base or the CA, and then instructs the GUI to update its display accordingly.
- The *rule base* contains a representation of the domain knowledge, for example, knowledge about an organisation's policies and procedures on bullying and harassment. Through dialogue with the CA, the rule base will gather information about the user's circumstances and will inform the controller which attributes and its associated value needs to capture in order to progress. Once all the information has been captured from the user, the rule base returns the advice to the controller using the attributes and their collective values.
- The *dialogue agent* undertakes dialogue with the user and has the ability to understand natural language queries, and formulate a response. The role of the agent is to capture information from the user about their personal circumstances and to answer questions about the specific domain. The CA obtains its own dialogue from a number of scripts, each representing a given context. Each script contains a series of rules and patterns which have been semi-automatically scripted by humans who have gained knowledge of the domain through the knowledge engineering process.
- The *graphical user interface* component manages the display and responds to the users requests either in the form of mouse click, button selection or natural language dialogue. The interface allows further conversation to take place e.g. it would allow the user to query the response given by the CA or ask any questions about the domain.

Case Study 1: Student Debt Advisor

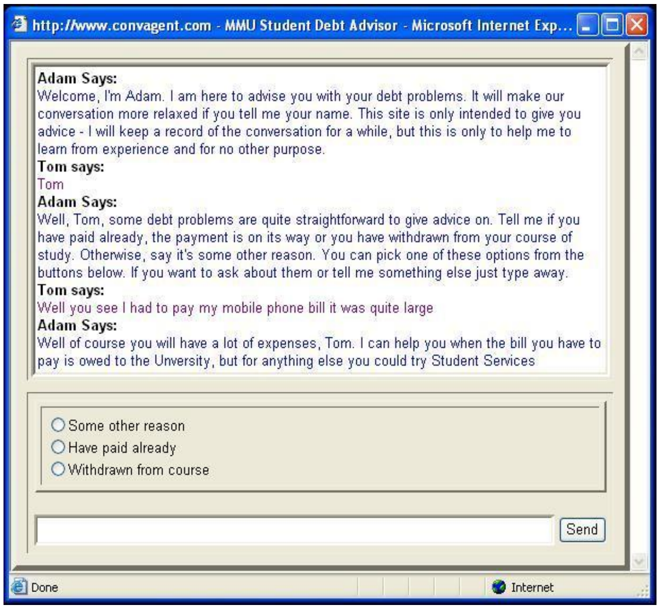
The first case study describes a CA called Adam which was developed by Convagent Ltd. Adam is a UK University Student Debt Advisor and is highly focused on providing advice to students by finding out why they have not paid their debts and offering advice on how they can find ways to pay them. Student debt is a growing concern in the UK with the average cost of a three-year University degree ranging from £39,000 to £45,000 (Qureshi, 2008). This includes not only variable University top up fees but basic living costs including rent, utility bills, travel and textbooks. Although UK home student can apply for student loans, bursaries and other financial help, the UK government has found that 65% of students are unaware of the financial help that they could receive (Qureshi, 2008).

The 'Adam' CA was designed to simulate the behaviour of student debt advisors within the university and give clear and consistent advice on what help was available to students who had debt problems. The student would then communicate with the "University Debt procedures" in every day English and receive guidance, ask questions and generally be led through the process as if by an expert human guide.

Student Debt Advisor Methodology

The student interacts with the agent by entering a natural language statement into an instant messaging style user interface, which is passed to a discourse manager for analysis. The discourse manager acts as a supervisor for the dialogue agent and 'thinks' about what sort of response to give back to the user. Any relevant information, such as the students name or how much they could pay towards their debt is extracted from the natural language statement and passed to a rule base which makes a decision on how to proceed in giving advice. The name is used to humanize the conversation and can be used in conjunction with a student id to pull back student financial account information. If no relevant information can be extracted then the conversational agent will engage in a dialogue to guide the students through a particular step in the process which is aimed at capturing a specific piece of information that will then allow the CA to provide further advice in solving the student's debt problem. If the student fails to cooperate with the conversational agent, after several attempts then the session will end.

The CA was designed to capture all the relevant information from the student's natural language dialogue in order to help them with their debt. Knowledge trees were used to represent the policies for student debt management within the organization. The CA, guided by the knowledge tree had awareness of the type of attributes that could be provided by the student, such as database attribute names, database values, numbers, dates, comparative terms such as "more than", "less than" etc. In order to achieve this, a different script was developed to identify the presence or absence of each possible relevant attribute and its value that would allow the CA to guide the user through the University student debt advisor process. An example of the dialog between a student 'Tom' and Adam can be seen in Figure 2.


http://www.convagent.com - MMU Student Debt Advisor - Microsoft Internet Exp...
Send
Done
Internet

Adam Says:
Welcome, I'm Adam. I am here to advise you with your debt problems. It will make our conversation more relaxed if you tell me your name. This site is only intended to give you advice - I will keep a record of the conversation for a while, but this is only to help me to learn from experience and for no other purpose.
Tom says:
Tom
Adam Says:
Well, Tom, some debt problems are quite straightforward to give advice on. Tell me if you have paid already, the payment is on its way or you have withdrawn from your course of study. Otherwise, say it's some other reason. You can pick one of these options from the buttons below. If you want to ask about them or tell me something else just type away.
Tom says:
Well you see I had to pay my mobile phone bill it was quite large
Adam Says:
Well of course you will have a lot of expenses, Tom. I can help you when the bill you have to pay is owed to the Unversity, but for anything else you could try Student Services

○ Some other reason
○ Have paid already
○ Withdrawn from course

Fig. 2. Sample interaction with 'Adam'

Evaluation of the Student Debt Advisor

The CA, Adam was subjected to four sets of tests, each comprising of 50 undergraduate students before going live. Students were given one of 5 scenarios which were designed from real life student circumstances of debt, such as the one shown in Figure 3, which illustrates one of the two most common reasons for students to call the University human helpline.

Students were asked to provide feedback of their experience of 'Adam', using a questionnaire. 80% of students were happy with the speed in which Adam elicited the responses; 70% of students were satisfied by the advice provided by Adam and 75% thought that Adam answered any additional queries that they had satisfactory. In the last part of the questionnaire, students were asked, "If you needed advice about your student debt problems, where would you go first?" The possible answers were

a) Use Adam instead of telephoning the University Finance Office
b) Use Adam instead of visiting the University Finance Office
c) Telephone the University Finance Office
d) Visit the University Finance Office
e) None of the above

Scenario 1. Tom
Tom is basically a well motivated student. He let himself get behind in his payments to the University but he is honest and he posted a cheque to the University as soon as he was able. Unfortunately the University issued the warning letter before Tom's cheque was logged in. Having gone to some pains to get the money together he now feels a bit indignant. He wants to let the University know he has paid (so that he is not chased any further), but he has gone from feeling guilty to gaining the moral high ground(so he feels) and he wants to let off steam. In playing the role of Tom we suggest that you do not let rip with a torrent of abuse initially, but to test Adam's response to abuse behaviour (modelled on the policies of counter staff) have a second go and swear at him.

Fig. 3 .Sample test scenario

47% of students stated that they would use Adam in instead of visiting the University Finance Office while 20% stated they would use Adam instead of telephoning the University Finance Office. From this initial testing phase, the majority of student's comments were positive. For example "He needs to be a bit friendly, I felt like I was seeing a real Advisor…. No wait that's a good thing!" However, the testing phase identified a number of areas where Adam could be improved such as help in additional areas, a better understanding of student life experiences and the use of slang and mobile text talk. These ideas were then incorporated into the ‘live’ version of Adam which is currently being run at the University. Access to Adam is through the University Finance web site (Adam, 2009). Examples of typical student debt scenarios can be found at www.convagent.com Further studies into its impact on dealing with student debt are on-going and feedback obtained from the CA log files enables ‘Adam’ to continue to learn and relate to life as a student.

Case Study 2: HR Bullying and Harrasment Advisor

The second case study will describe how a conversational agent was used to act as an advisor for employees on ‘Bullying and Harassment in the Workplace’ policy and procedures in a large organisation (Latham, 2007). In the UK, no single piece of legislation addresses the problem of bullying and harassment (Latham, 2007). Rather, organisations are required to implement a number of different laws which protect employees from harassment due to a number of different causes. Implementing such legislation has led organisations to develop comprehensive policies, often resulting in large and complex policy documents. This type of policy will often require additional training and guidance support for members of the organisation wishing to understand and follow the process for reporting bullying and harassment. The high support cost and fairly static nature of such policies means that they are suitable for automation using a conversational agent. This would allow anonymous advice to be available 24 hours a day and ensuring that the advice given is consistent, appropriate and valid. In addition, the employee has the ability to query any advice given and ask for further explanations. In developing the conversational agent, knowledge engineering was used to elicit the main questions in relation to bullying and harassment asked by employees within the organisation (Latham, 2007). These were identified as:

1. What counts as bullying and harassment?
2. How do I report bullying or harassment?
3. What can I do if I am not being taken seriously?
4. What can I do if I am being victimised?

Methodology and System Architecture

A rule base was then used to structure the bullying and harassment domain. The main purpose of using a rule base for knowledge representation is that its structure is similar to the way in which humans represent knowledge. Figure 4 shows a portion of the rule base for structuring knowledge about the ‘I am being victimized’ option. Each context consisted of a number of hierarchically organized rules where each rule possesses a list of patterns and an associated response. An example rule for dealing with an employee who expresses confusion in natural language about whether they are actually being bullied is shown below:

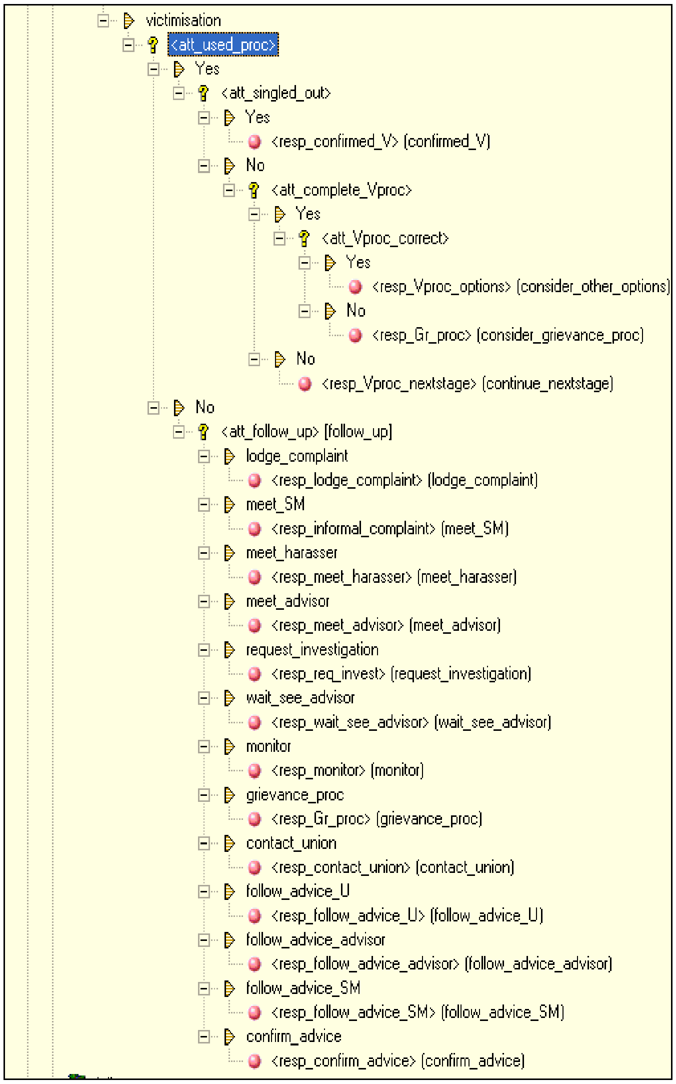
```
<Rule_04>
  .   *<confused-0>*
  .   *<confusing-0>*
  .   *<sure-neg-0>*
  .   *<sure-neg-1>*
  .   *help*
  .   *not *<understand-0>*

r: How can I help you
```

where p is the strength of the pattern and r the response.

Patterns can also contain wildcard elements “*” which will match with one or more consecutive characters. In addition, the macro “<confused-0>” enables the scripter to incorporate stock patterns into a rule.

Through natural language dialogue with the user, the conversational agent would look for each possible relevant attribute and any associated value in the users input in order to determine what rule in the rule base to fire and hence what advice to give. When user input is passed from the user interface to the conversational agent, all the different scripts will be called in turn. Each script will determine whether a specific attribute and its associated value is present in the input, and if it is, capture it and assign it to a set of variables. These variables will then be passed on to the rule base in order to determine the direction of the conversation and the type of response the conversational agent should provide to the user.

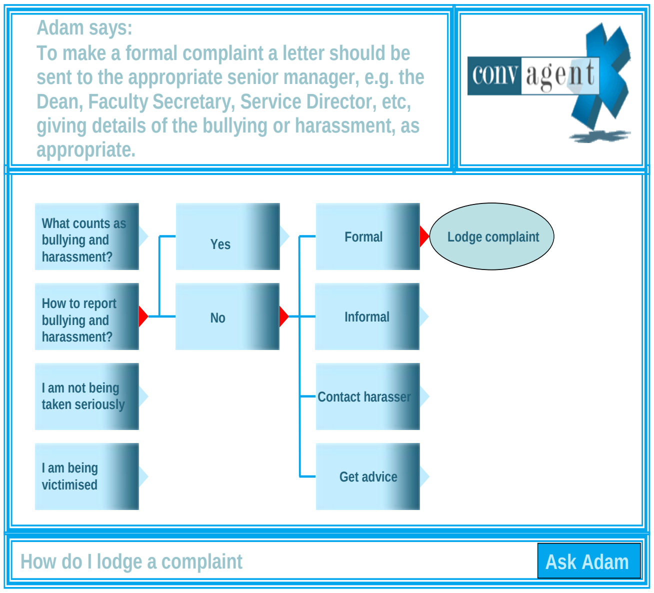


```
victimisation
  <att_used_proc>
    Yes
      <att_singled_out>
        Yes
          <resp_confirmed_V> (confirmed_V)
        No
          <att_complete_Vproc>
            Yes
              <att_Vproc_correct>
                Yes
                  <resp_Vproc_options> (consider_other_options)
                No
                  <resp_Gr_proc> (consider_grievance_proc)
            No
              <resp_Vproc_nextstage> (continue_nextstage)
    No
      <att_follow_up> (follow_up)
        lodge_complaint
          <resp_lodge_complaint> (lodge_complaint)
        meet_SM
          <resp_informal_complaint> (meet_SM)
        meet_harasser
          <resp_meet_harasser> (meet_harasser)
        meet_advisor
          <resp_meet_advisor> (meet_advisor)
        request_investigation
          <resp_req_invest> (request_investigation)
        wait_see_advisor
          <resp_wait_see_advisor> (wait_see_advisor)
        monitor
          <resp_monitor> (monitor)
        grievance_proc
          <resp_Gr_proc> (grievance_proc)
        contact_union
          <resp_contact_union> (contact_union)
        follow_advice_U
          <resp_follow_advice_U> (follow_advice_U)
        follow_advice_advisor
          <resp_follow_advice_advisor> (follow_advice_advisor)
        follow_advice_SM
          <resp_follow_advice_SM> (follow_advice_SM)
        confirm_advice
          <resp_confirm_advice> (confirm_advice)
```

Fig. 4. Portion of rule base

Figure 5 shows an example of a dialogue interaction between an employee and the bullying and harassment advisor 'Adam'. The dynamic interface presents the user with three methods of communication: graphical (clickable buttons) or through natural language dialogue or in a mixed mode. The interface was designed so that more experienced users were able to select options by simply clicking on their choices, whilst concurrently or independently, natural language dialogue can also be entered in order to discuss a persons individual circumstances, explain concepts and offer advice at any stage in the process. The dialogue between the User and Adam, to reach a point in the conversation as shown in Figure 5 is as follows:

User: I think I may be being bullied, how do I report it.
Adam: Have you taken any action yet?
User: No
Adam: What action would you like to take?
User: Definetly formal
Adam: To make a formal complaint a letter should be sent to the appropriate senior manager, e.g. the Dean, Faculty Secretary, Service Director, etc, giving details of the bullying or harassment, as appropriate.
User: How do I lodge a complaint?....................



Adam says:
To make a formal complaint a letter should be sent to the appropriate senior manager, e.g. the Dean, Faculty Secretary, Service Director, etc, giving details of the bullying or harassment, as appropriate.

How do I lodge a complaint

conv agent
What counts as bullying and harassment?
How to report bullying and harassment?
I am not being taken seriously
I am being victimised
Yes
No
Formal
Informal
Contact harasser
Get advice
Lodge complaint
Ask Adam

Fig. 5. Bullying and Harassment Advisor

In addition, the system has a built in domain spell checker so that correct responses can be given even if the user miss-spells words e.g. "Definetly".

Evaluation of the HR Bullying and Harassment Advisor

A representative group of 30 employees' were selected to evaluate the CA. The group of 30 employees worked in different roles and for diverse companies, aged between 25 and 45. The group included both male and female members from administrative, academic and managerial backgrounds. A scenario of possible sexual harassment was developed along with a questionnaire to record feedback. The group were asked to read the scenario and then complete two tasks within the system. Users then completed an electronic evaluation questionnaire anonymously. Ease of use is critical to the success of the conversational agent in supporting a Bullying and Harassment policy. Users felt that the system was intuitive and easy to use, giving high scores to ease of navigation and the ability to find information. On the whole, scores were high and the group found the system understandable and intuitive to use. 94% of users indicated that they had found the advice they sought without difficulty and one user commented that they "*did not need to 'learn' to use the advisor*". This clearly is a very important benefit as users who seek to use this type of system do so because they want the same level of advice as provided by a human, without being constrained due to their technical ability. As the Bullying and Harassment Advisor covers an extremely sensitive domain, it is not currently deployed as a live system. Comprehensive testing and evaluation of the advisor is currently being undertaken with both domain experts and potential users.

Issues in Developing Conversational Agents

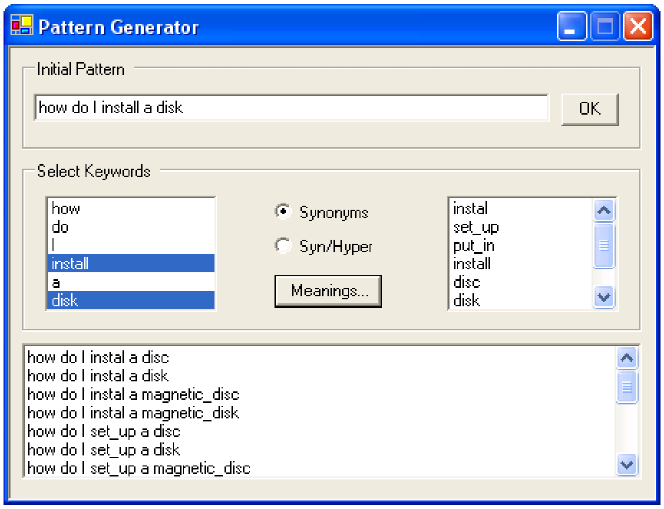
Whilst the two cases described in this paper highlight the clear benefits of using conversational agents as human advisors, there are several research issues that need to be addressed which surround the development of all conversational agents. There are four main problems associated with the text based conversational agents:

- General (breadth) verses specific knowledge (depth). By limiting a CA to a specific domain, extensive in-depth knowledge can be learned as the CA encapsulates the knowledge of the experts through the knowledge engineering process. This strategy allows the CA to provide clear and consistent advice about the domain whilst providing very general responses to any questions outside the domain. It will also be able to simulate some aspects of human behaviour in providing responses to general day to day conversation. It will however fail to have any general knowledge for example on recent media events.
- Scripting and Authoring: Capturing the domain knowledge and then scripting a CA is a labour-intensive process. Changes in the environment and/or in the knowledge would effectively mean that new scripts would need to be created meaning the maintenance of such systems is high and the process of incorporating new scripts could lead to conflicting rules in the rule base.
- Speed. Can the agent operate fast enough to maintain a consistent dialogue with humans? Scalability is an issue in large organisations where large number of agents are in use simultaneously and interacting with the same rule base. This can be overcome by designing each agent to act autonomously across the network.
- Modelling human behaviour. One of the main differences between a human and a CA advisor is the ability to exhibit social behaviour. Intelligence about a domain can be learned but the application of this intelligence may depend on the social situation in which it occurs. In order for the CA to give a response that was reactive to human behaviour, the CA would have to have social awareness not only of its self but also of others at each stage within the conversation.

The following sections will review some of the current research, mainly focusing on the two key areas of script generation and evaluation, which is currently being undertaken to try and tackle some on the above problems.

Script Generation

Traditionally, each new CA will require a vast quantity of script files to be created by humans which is a very labour intensive process. Whilst it is possible to re-use general scripts, for example to capture the name of a user, the majority of scripts will be domain specific. The solution is to either find some way to automatically generate the scripts themselves or to move away from the pattern matching ideology of scripting in CA's. Webster et al (Webster et al, 2007), have developed an Automatic Pattern Generation Tool which utilizes WorldNet (Miller, 1995) to automatically generate lists of patterns based on key terms about the domain and formulates them into rules inside a script. WorldNet is a lexical database which stores English nouns, verbs, adjectives and adverbs which are organized into synonym sets or *synsets*, each representing one underlying lexical concept. Noun synsets are related to each other through *hypernymy* (generalization), *hyponymy* (specialization), *holonymy* (whole of) and *meronymy* (part of) relations. Of these, (*hypernymy*, *hyponymy*) and (*meronymy*, *holonymy*) are complementary pairs (Miller, 1995). From the computational perspective, it is a massive and well-structured database, with thousands of words and meanings organized into a semantic network. The Automatic Pattern Generation Tool uses only the concepts of synonymy and hyponymy in order to generate patterns. The tool uses patterns from WorldNet which are semantically similar to those entered by the user. Figure 6 shows a sample of the reconstructed rules and established patterns that have been generated for the sentence "how do I install a disk". Whilst the tool goes some way in speeding up the scripting process it still requires a substantial degree of human interaction



how do I install a disk

how
do
I
install
a
disk

● Synonyms
○ Syn/Hyper

instal
set_up
put_in
install
disc
disk

how do I instal a disc
how do I instal a disk
how do I instal a magnetic_disc
how do I instal a magnetic_disk
how do I set_up a disc
how do I set_up a disk
how do I set_up a magnetic_disc

Fig. 6 Reconstructed sentences from the initial pattern

An alternative approach involves moving away from the use of pattern matching in CA's. O'Shea et al (O'Shea et al a, 2008; O'Shea et al b, 2008;) have recently proposed a novel approach to the scripting of Conversational Agents (CA) through the use of sentence similarity measures. Sentence Similarity Based on Semantic Nets and Corpus Statistics (Li et al, 2002; Li et al, 2003; Li et al, 2006) is a measure that focuses directly on computing the similarity between very short texts of sentence length. In the proposed methodology, sentence similarity measures are employed to examine semantics rather than structural patterns of sentences. Initial results have indicated a substantial reduction in the CA's rule base as scripted patterns in all rules are instead replaced with one or two natural language sentences. This reduces the overhead of the initial scripting process and the future maintenance of the CA for any given domain. Hence, the labour-intensive process of scripting could be reduced radically.

Evaluation of conversational agents

Evaluating text based conversational agents is difficult to achieve as there is no clear scientific benchmark. Typically evaluations have been based on variants of the Turing Test. Early evaluation methodologies such as PARADISE (Walker et al, 2001) were designed for speech-based conversational agents where the overall goal in the conversation was user satisfaction. Formative evaluation has been carried out by Sanders and Scholtz for the proposed DARPA Communicator (Sanders and Scholtz, 2000). Each conversational agent was evaluated on criteria such as correct task completion, the cost of the completing the task and the quality of interaction between the system and the end user. This work led to Sanders and Scholtz proposing eight general heuristics which were validated through correlations with user satisfaction ratings and with quantitative metrics. These heuristics were (Sanders and Scholtz, 2000):

1. The systems functionality and use is clear to users
2. The system takes active responsibility for repair when possible
3. The system uses clear, concise and consistent language
4. The system follows the dialogue conventions of natural human dialogues.
5. The system gives prompt coherent feedback
6. The system correctly handles answers that give more information than requested.
7. The system detects conflicting input from the user, notifies the user, and actively clarifies or repairs the problem
8. The system supports users with different levels of expertise.

The proposed metrics were constructed from a number of low level counts which were taken from the log files of the conversation and from user evaluation which is subjective. A scientific approach to short text similarity evaluation has been proposed by O'Shea et al (O'Shea et al b, 2008) who has developed a benchmark data set of 65 sentence pairs with human-derived similarity ratings. This data set is the first of its kind, specifically developed to evaluate short text sentence similarity measures and will be a viable measure in the evaluation of text based CA's.

Social behaviour

The majority of current research has centered on the incorporation of specific human behaviours in embodied conversational agents. Two key areas of development are believability and the social interface with human beings. The concept of believability examines how an embodied agent may encapsulate emotion, exhibit a strong personality and convey the "illusion of life" (Becker et al, 2007; Bickmore and Picard. 2005; Bickmore and Schulman 2007)). The social interface includes the ability of the agent to cooperate with other agents in a social context through dialog and become "socially aware". A major issue is the lack of qualitative and quantifiable evaluation of behaviour in such embodied conversational agents. It has yet to be established whether or not such human behaviours can be solely learnt and applied within text based conversational agents (Bickmore and Giorgino,2006).

Conclusions and Further Work

This paper has introduced the concept of conversational agents acting as human advisors to an organisations policies and procedures. A novel methodology for constructing text based conversational agent advisors has been described and two case studies which employ this methodology were introduced and evaluated. The first system, the Student Debt Advisor is currently in use at Manchester Metropolitan University and is consistently undergoing evaluation and review. The second, the HR Bullying and Harassment Advisor is currently being evaluated by domain experts and potential users.. Further systems employing the proposed methodology are also undergoing development, including a Staff Travel Advisor. Finally, this paper concluded by examining a number of research issues in designing text based conversational agents as human advisors have been highlighted and areas of current research were discussed. Significant further work is required, firstly, in developing the first conversational agent solely based on sentence similarity measures rather than the pattern matching approach and secondly, in evaluating its performance scientifically, using a bench mark sentence data set.

Acknowledgment

The authors wish to thank Convagent Ltd for the use of their conversational agent engine for use within this research.

References

ADAM, Student Debt Advisor, Available: http://www.fin.mmu.ac.uk/. Accessed: 06 May 2009.

A.L.I.C.E. AI Foundation, Inc. 2007, The A.L.I.C.E. Artificial Intelligence Foundation, Available: http://www.alicebot.org/, accessed viewed: 07 March 2009

Becker, C., Kopp, S., & Wachsmuth, I. Why emotions should be integrated into conversational agents. Conversational Informatics: An Engineering Approach (pp. 49-68). Wiley, 2007

Bickmore, T., Giorgino, T.: Health dialog systems for patients and consumers. J. Biomed.Inform. 39(5), pp 556–571, 2006

Bickmore, T, Pickard, R. Establishing and maintaining long-term human-computer relationships. ACM Trans. Comput.-Hum. Interact. 12(2): pp293-327., 2005

Bickmore, T. Schulman, S. Practical approaches to comforting users with relational agents. CHI Extended Abstracts 2007: 2291-2296, 2007

Colby, K. Artificial Paranoia: A Computer Simulation of Paranoid Process, Pergamon Press., New York, 1975.

Cassell J, Sullivan, J. Prevost. S, Churchill, E, Embodied Conversational Agents, MIT press, 2000.

Iacobelli, F. & Cassell, J. “Ethnic Identity and Engagement in Embodied Conversational Agents” Proceedings of Intelligent Virtual Agents (IVA), Paris, France, 2007

Cassell, J., Gill, A. & Tepper, P. Coordination in Conversation and Rapport. Workshop on Embodied Natural Language, Association for Computational Linguistics. Prague, 2007

Convagent Ltd, Available: www.convagent.com. Accessed: March 7, 2009

Latham, A. An Expert System for a Bullying and Harassment Policy, MSC thesis, Manchester Metropolitan University, 2007

O’Shea, K. Bandar, Z Crockett, K. A Novel Approach for Scripting Conversational Agents using Sentence Similarity Measures. Technical Report Manchester Metropolitan University, 2008.

O'Shea, J., Bandar, Z., Crockett, K., McLean, D., A Comparative Study of Two Short Text Semantic Similarity Measures, Lecture Notes in Artificial Intelligence, LNAI, Vol.4953,pp172 Springer,2008. Li Y, Bandar Z, McLean, D. Measuring Semantic Similarity between Words Using Lexical Knowledge and Neural Networks. LNCS, Springer-Verlag, pp 481-486, 2002.

Li, Y. Bandar, Z. O’Shea, J. Mclean, D. Crockett, K. Sentence Similarity Based on Semantic Nets and Corpus Statistics, IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 8, pp. 1138-1150, 2006.

Li, Y. Bandar, Z. McLean, D. An approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE Transactions on Knowledge and Data Engineering., Vol 15:4, pp:871-881, July/August 2003.

Massaro, D.W., Cohen, M. M., Beskow, J., & Cole, R. A. “Developing and evaluating conversational agents”, In J. Cassell, J. Sullivan, S. Prevost, & E. Churchill (Eds.) Embodied conversational agents, MIT Press, Cambridge, 2000.

Massaro, D. A framework for evaluating multimodal integration by humans and a role for embodied conversational agents. ICMI, pp 24-31, 2004

Mauldin, M. Chatterbots, Tinymuds, and the Turing Test: Entering The Loebner Prize Competition, AAAI 1994

Michie, D. Sammut C, Infochat Scripter's Manual, Convagent Ltd, Manchester, UK, 2001.

Miller, G “WordNet: A Lexical Database for English”, Comm. ACM, Vol. 38, no. 11, pp. 39-41, 1995

Owda, M. Crockett, K., Bandar, Z, Conversation-Based Natural Language Interface to Relational Database, IEEE International Conferences on Web Intelligence and Intelligent Agent Technology, pp 363-367, 2007

Popescu, A. Etzioni, O. Kautz, H 'Towards a Theory of Natural Language Interfaces to Databases', in Proceedings of the 8th International Conference on Intelligent User Interfaces, Miami, Florida, pp 149-157, 2003

Pudner, K. Crockett, K. Bandar, Z. An Intelligent Conversational Agent Approach to Extracting Queries from Natural Language, World Congress on Engineering, International Conference of Data Mining and Knowledge Engineering, pp 305-310, 2007

Qureshi, H. Students feel the pinch of top-up fees. Available: http://www.guardian.co.uk/money/2008/feb/03/studentfinance.edu c ation Accessed 12/2/08.

Sanders, G. A., and Scholtz, J., Measurement and Evaluation of Embodied Conversational Agents, 2000 Cassell, J., Sullivan, J., Prevost, S., and Churchill, E., eds., Embodied Conversational Agents, 2000

Turing, A. M., 1950 Computing Machinery and Intelligence, Mind, New Series, V59, issue 236, pp 433-460, 1950.

VirtuOz 2009, VirtuOz, conversational agents, Available: http://www.virtuoz.com/en/?gclid=CNX-1bDbiY4CFRAFEgodeXC2Ew accessed: March 7, 2009

Walker, M. Passonneau, R. Boland, J Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems. In Proc. of the Meeting of the Association of Computational Linguistics, ACL 2001, 2001.

Webster, R. Crockett, K, Bandar, Z. Automatic Script Generation Tool, MSc thesis, Manchester Metropolitan University, 2007

Weizenbaum, J. "ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine," Communications of the Association for Computing Machinery 9, pp36-45, 1996

Xiao, J. Empirical Studies on Embodied Conversational Agents, GIT Publishing, 2006. Available: http://etd.gatech.edu/theses/available/etd-09222006-140611/, Accessed: March 7, 2009

An Integrated Authoring Tool for Tactical Questioning Dialogue Systems

Sudeep Gandhe and **Nicolle Whitman** and **David Traum** and **Ron Artstein**
Institute for Creative Technologies
13274 Fiji way, Marina del Rey, CA 90292.
<*lastname*>@ict.usc.edu

Abstract

We present an integrated authoring tool for rapid prototyping of dialogue systems for virtual humans taking part in tactical questioning simulations. The tool helps domain experts, who may have little or no knowledge of linguistics or computer science, to build virtual characters that can play the role of the interviewee. Working in a top-down fashion, the authoring process begins with specifying a domain of knowledge for the character; the authoring tool generates all relevant dialogue acts and allows authors to assign the language that will be used to refer to the domain elements. The authoring tool can also be used to manipulate some aspects of the dialogue strategies employed by the virtual characters, and it also supports re-using some of the authored content across different characters.

Introduction

Tactical Questioning dialogues are those in which small-unit military personnel, usually on patrol, hold conversations with individuals to produce information of military value (Army 2006). Building Tactical Questioning characters that can play the role of a person being questioned has been an on-going project at Institute for Createive Technologies. The simulation training environment can be used to train military personnel in how to conduct such dialogues. The project has evolved through many different architectures for dialogue systems (Traum et al. 2008). Gandhe et al. (2008) provide description of the latest architecture for this tactical questioning dialogue system.

These tactical questioning dialogues are different from typical question answering dialogues in that they can be non-cooperative at times. The character may answer some of the questions by the interviewer in a cooperative manner but some other questions which are of a more sensitive nature may need more coercion from the interviewer. Some of the strategies used by interviewers include building rapport with the character, addressing their concerns, promising to do certain actions in their favor or pointing out the effects of non-cooperation.

Traditionally, one step in the development life cycle for a dialogue system is to build a corpus of in-domain human-human dialogues through roleplays or Wizard of Oz sessions; this is the starting point for specifying the domain. Corpus collection can be costly and time-consuming. If the domain of interaction is relatively simple and can be authored *consistently* and *completely* by a scenario designer, the collection of dialogue corpora can be bypassed. Here *consistency* refers to generating only the valid dialogue acts that can be correctly handled by the dialogue manager and *completeness* refers to generating all dialogue acts that are relevant with respect to the character's domain knowledge and associating all of these to corresponding surface text.

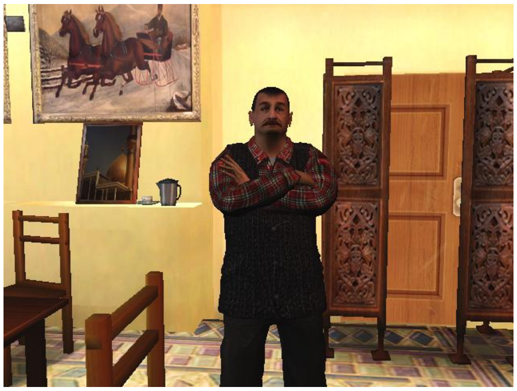


Figure 1: Hassan – A virtual human for Tactical Questioning

We have implemented a character named Hassan (see Figure 1), who is being questioned about illegal tax collections at a local marketplace. We will use this domain for most of the examples in this paper.

In the next section, we review some of the existing authoring tools for building dialogue systems. We then list the required features an authoring tool should provide and explain our design decisions for the tool. Next, we describe the tool starting with how the domain knowledge is specified. The subsequent section explains how dialogue acts are automatically generated based on the domain knowledge and how the dialogue manager functions at the dialogue act level. It is followed by the discussion of surface text authoring and how authored content can be re-used across multiple characters. We present a preliminary evaluation of the tool and conclude by discussing avenues for future improvements.

Related Work

Many toolkits and authoring environments have been developed for building dialogue systems. Rapid Application Developer from CSLU toolkit (Sutton et al. 1998) allowed designers to build dialogue systems employing finite state dialogue models. The authoring environment was accessible by non-experts and allowed building systems that could conduct simple directed dialogues. Our tactical questioning dialogue system is mainly reactive but allows for some initiative for simple negotiations. It can be cast into finite state models augmented with information state. Since our virtual human system engages the trainee in a natural conversation, the input from the user is free-form and is more challenging for the NLU.

There have been several commercial dialogue building solutions based on VoiceXML, which allows for a form-based dialogue management. RavenClaw (Bohus and Rudnicky 2003) is another dialogue architecture where designers can specify hierarchical domain task specification. The dialogue management builds on top of agenda based dialogue management technique (Xu and Rudnicky 2000). Although this architecture has been successfully used for building multiple dialogue systems, it is most suited for task-oriented dialogue systems and using it requires considerable expertise in programming and design of dialogue systems. Other dialogue system architectures such as TrindiKit (Larsson et al. 2004) or Midiki (MITRE 2005), which use information state based dialogue modeling (Traum and Larsson 2003) have the same issue. These systems require considerable knowledge of the dialogue theories and software development.

There have been some efforts in the area of tutorial dialogue systems that concentrate on building authoring tools which can be used by non-experts for rapidly building a dialogue system. TuTalk (Jordan et al. 2007) is one such system. TuTalk authoring tool allows tutorial system researchers who may not have expertise in the dialogue system design to rapidly prototype dialogue systems and experiment with different ideas. Our tactical questioning project has a similar requirement. The TuTalk authoring tool allows authoring of initiation-response pairs along with many features suitable for tutorial dialogue systems.

Our initial efforts in providing authoring tools for tactical questioning were along the same lines. Designers were allowed to author questions and the corresponding answers (Leuski et al. 2006). Although this works very well for simple question answering systems, it suffers from the inability to maintain coherence over sequences of utterances greater in length than two. We need this ability to engage in simple negotiation dialogues. We follow an approach similar to TuTalk in designing an authoring tool that is specialized for a specific genre of dialogue viz. tactical questioning and allows authoring by non-experts.

Requirements

One of the requirements for the tactical questioning project is to allow subject matter experts to rapidly build different scenarios within the same tactical questioning framework. Moreover, these authors should not require any expertise in linguistics or computer science. For these reasons, we designed a simple schema for specifying the domain knowledge which is easily understandable. The authoring process starts with the domain knowledge construction which is done with the help of our authoring tool (see Figure 2). The authoring tool automatically constructs all relevant dialogue acts that are used by the dialogue manager. The tool also allows direct linking of these acts to surface text of the utterances for training NLU and NLG.

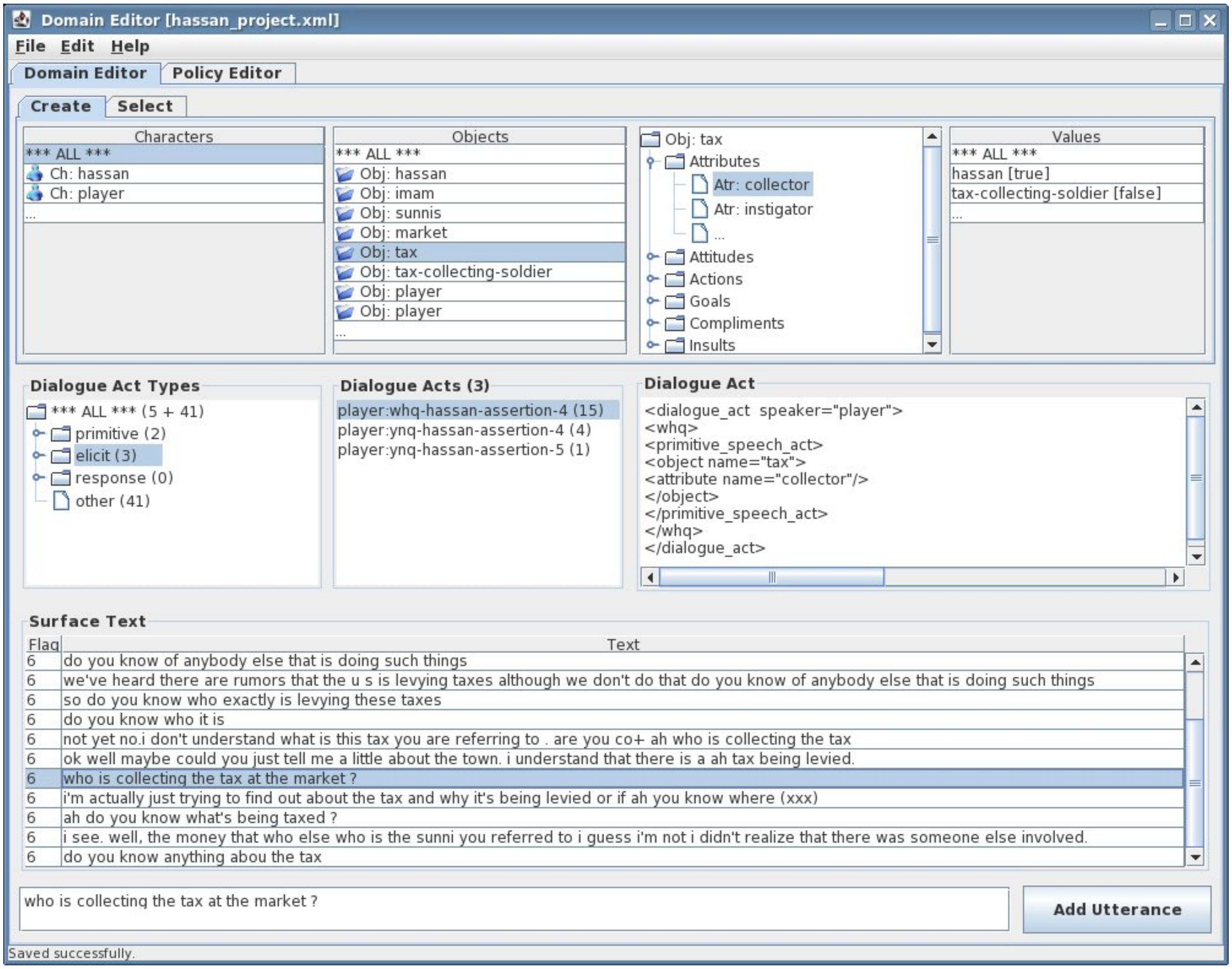
Although Tactical Questioning dialogues are mainly question-answering dialogues, we need the ability to model simple negotiations over when to release certain sensitive information. The dialogue manager maintains a model of emotions and compliance which are updated as the dialogue proceeds. In compliant mode, the character may elicit certain offers from the interviewer before answering questions regarding the sensitive information. Whereas in adversarial mode, the character may choose to lie in response to these questions. We need to allow the scenario authors to mark certain information elements as sensitive and modify some of the policies regarding when to release this information.

There are cases where we would like to build several characters that can be questioned about the same incident. E.g. Multiple witnesses of a shooting incident at the marketplace will have a considerable overlap in their domain knowledge. One of the requirements for the authoring tool is the ability to re-use the existing authored content across different characters. Our tool allows for such re-use of the domain knowledge along with all the dialogue acts and the language associated with it.

Figure 2 shows a screenshot of our authoring tool. It has three horizontal panels. The topmost panel is used for editing the domain knowledge level. The middle one allows authors to view all dialogue acts and select one of them. The bottom panel allows editing of the surface text corresponding to the chosen dialogue act.

Domain Knowledge Level

Domain knowledge is created as a four level hierarchy. The highest level is the *characters*, the conversational participants in the domain, who can be speakers and addressees of utterances and dialogue acts. In the Hassan domain there are two characters viz. the trainee (called player) and Hassan. Each character knows about a set of *objects*. These objects can be of different types such as person (imam), location (market) or abstract concept (tax). Each object can be further described by *attributes*. Finally, attributes can take on *values*, some of which can be marked false – to be used as lies. A basic proposition is a triple <object,attribute,value>. Queries for the value field of a such propositions form the basis for questions. Objects of type person can also have representations of the actions they can perform (e.g. offers, threats, admissions), their *goals*, and their *attitudes* toward other objects. Actions and *goals* are not further specified with values. *Attitudes* are used in a similar fashion to attributes. Currently *attitudes* and *goals* are used as talking points only. In future, we plan to connect *goals* with actions and other domain knowledge. These additional aspects are



Characters
*** ALL ***
Ch: hassan
Ch: player
...

Objects
*** ALL ***
Obj: hassan
Obj: imam
Obj: sunnis
Obj: market
Obj: tax
Obj: tax-collecting-soldier
Obj: player
Obj: player
...

Obj: tax
- Attributes
  - Atr: collector
  - Atr: instigator
  - ...
- Attitudes
- Actions
- Goals
- Compliments
- Insults

Values
*** ALL ***
hassan [true]
tax-collecting-soldier [false]
...

Dialogue Act Types

- *** ALL *** (5 + 41)
  - primitive (2)
  - elicit (3)
  - response (0)
  - other (41)

Dialogue Acts (3)

player:whq-hassan-assertion-4 (15)
player:ynq-hassan-assertion-4 (4)
player:ynq-hassan-assertion-5 (1)

Dialogue Act

```
<dialogue_act speaker="player">
<whq>
<primitive_speech_act>
<object name="tax">
<attribute name="collector"/>
</object>
</primitive_speech_act>
</whq>
</dialogue_act>
```

Surface Text

Flag	Text
6	do you know of anybody else that is doing such things
6	we've heard there are rumors that the u s is levying taxes although we don't do that do you know of anybody else that is doing such things
6	so do you know who exactly is levying these taxes
6	do you know who it is
6	not yet no.i don't understand what is this tax you are referring to . are you co+ ah who is collecting the tax
6	ok well maybe could you just tell me a little about the town. i understand that there is a ah tax being levied.
6	who is collecting the tax at the market ?
6	i'm actually just trying to find out about the tax and why it's being levied or if ah you know where (xxx)
6	ah do you know what's being taxed ?
6	i see. well, the money that who else who is the sunni you referred to i guess i'm not i didn't realize that there was someone else involved.
6	do you know anything abou the tax

who is collecting the tax at the market ?

Figure 2: A tool for designing the domain, dialogue acts and the utterances that map to those dialogue acts.

used to create policies for how to engage in questioning dialogues. Another aspect crucial for tactical questioning is social behavior for building rapport, oriented more generally to the characters rather than specific objects. We have sets of these kinds of content, including *compliments* and *insults*.

We use a simple XML representation of these domain knowledge aspects, for ease of use across modules of the system at both domain specification and run-time. Figure 3 shows parts of the domain specification used for the Hassan scenario, including an attribute and an action for Hassan, and both a true and false (i.e. lie) value for an attribute about the tax.

The topmost panel of the authoring tool (see Figure 2) shows the domain creation aspects, where, moving from left to right, authors can add or delete characters, objects, attributes (or other object contents) and values. The tool automatically constructs XML like in Figure 3. This top most section is also used to filter the set of dialogue acts shown in the middle panel of the GUI.

Dialogue Level

Once the domain is defined, it needs to be linked up with the language that will be used to refer to it. Dialogue acts form the middle level in this link, having domain aspects as their contents and being identified directly as the interpretations of language utterances.

Dialogue Acts

Our dialogue manager reasons about several standard types of dialogue acts, including *assertions*, *yn-questions*, *wh-questions*, *offers*, *threats*, *compliments* and *insults*. Following Core and Allen (1997) we have dialogue acts with forward-function – *elicitations* and with backward-function – *responses* for most of the acts. Figure 4 shows our XML representation of some of these acts, which contain a speaker (one of the characters), an act-type, and contents.

All dialogue acts are automatically created from the domain representation as per Algorithm 1. E.g. all <object,attribute,value> triples known by a character can

```
<domain name="hassan">
  <character name="hassan">
    <object name="hassan" type="person">
      <attribute name="role">
        <value>middle-man</value>
      </attribute>
      <actions>
        <offer name="cooperate"/>
      </actions>
    </object>
    <object name="tax" type="abstract">
      <attribute name="collector">
        <value>hassan</value>
        <value isTruth="false">
          tax-collecting-soldier
        </value>
      </attribute>
      ...
```

Figure 3: Aspects of the *Hassan* domain

hassan.assert

```
<dialogue_act speaker="hassan">
  <primitive_speech_act>
    <assertion>
      <object name="tax">
        <attribute name="collector">
          <value>hassan</value>
        </attribute>
      </object>
    </assertion>
  </primitive_speech_act>
</dialogue_act>
```

Indeed, you might say that I collect the taxes.

player.offer

```
<dialogue_act speaker="player">
  <primitive_speech_act>
    <offer name="give-money"/>
  </primitive_speech_act>
</dialogue_act>
```

We can offer you financial reward.

hassan.elicit-offer

```
<dialogue_act speaker="hassan">
  <elicit>
    <primitive_speech_act>
      <offer name="give-money"/>
    </primitive_speech_act>
  </elicit>
</dialogue_act>
```

I might tell you what you want if there was something in it for me.

Figure 4: Sample dialogue acts automatically generated from the *Hassan* domain along with example utterances.

Algorithm 1 Generation of dialogue acts from domain

```
for all $speaker \in$ characters do
  /* Primitive dialogue acts */
  for all $obj \in$ objects under $speaker$ do
    ADD assertions ($speaker$, $obj$, $atr$, $val$)
    ADD attitudes ($speaker$, $obj$, $atd$, $val$)
    ADD actions ($speaker$, $obj$, $act$)
    ADD goals ($speaker$, $obj$, $goal$)
    ADD compliments ($speaker$, $obj$, $compl$)
    ADD insults ($speaker$, $obj$, $insult$)
    ADD groundingDAs ($speaker$, $obj$)
  end for
  /* Dialogue acts that relate to other characters */
  for all $char' \in (\text{characters} \setminus speaker)$ do
    for all $obj' \in$ objects under $char'$ do
      /* Forward-looking dialogue acts */
      ADD whq ($speaker$, $obj'$, $atr'$)
      ADD ynq ($speaker$, $obj'$, $atr'$, $val'$)
      ADD elicit-action ($speaker$, $obj'$, $act'$)
      /* Backward-looking dialogue acts */
      ADD response-action ($speaker$, $obj'$, $act'$)
      ADD response-compl ($speaker$, $obj'$, $compl'$)
      ADD response-insult ($speaker$, $obj'$, $insult'$)
      ADD groundingDAs ($speaker$, $obj'$)
    end for
  end for
  /* Generic dialogue acts */
  ADD greetings, closings, accept, reject, refuse-answer,
  ack, offtopic, ...
end for
```

serve as the contents of an *assert* with that character as the speaker. Likewise, any <object,attribute> pair known by another character can be queried with a *wh-question* addressed to that character. We also generate some generic dialogue acts that are customary in human-human conversations like *greeting* and *closing*, that are not tied to any specific domain content. Grounding acts like *repeat-back*, *request-repair* are also generated. *Offtopic* is a special dialogue act specifically designed to handle out-of-domain dialogue acts from the player. The Hassan domain has 102 dialogue acts with Hassan as speaker and 108 dialogue acts with player as the speaker.

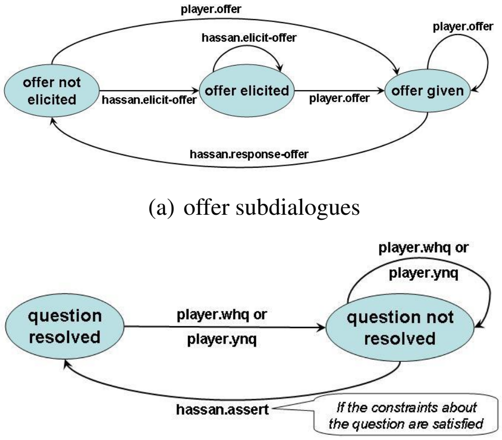
The middle panel of the authoring tool shown in Figure 2 allows selection from among the full set of dialogue acts. The left pane allows selection of the type of dialogue act; the middle pane lets one select individual dialogue acts; the right pane shows the full XML content of the dialogue act. In future, instead of showing a dialogue act with XML representation, we plan to use pseudo-natural language – possibly generated using templates. E.g. A template like "*Attribute* of *Object* is *Value*" for *assert* dialogue act type.

Dialogue Manager

Previous dialogue manager for tactical questioning characters like Hassan (Roque and Traum 2007) made use of hand-authored rules for tracking affective variables and offers and threats made. It used these to compute a *compliance level*, which would dictate how the character would respond. There were three possible compliance levels – adversarial, reticent and compliant. The system's response was determined using text-to-text mappings. That architecture required developers to specify complete input-text to output-text mappings for all three compliance levels. But this architecture could not handle the dependencies between utterances that go beyond just the adjacent ones.

In order to handle such dependencies and to reason at a more abstract level, the new dialogue architecture makes use of dialogue acts and the domain content. The dialogue manager used in this architecture is based on the information state model (Traum and Larsson 2003). The information-state is in part based on conversational game theory (Lewin 2000). The main responsibilities of the dialogue manager are to update the information state of the dialogue and use it to select the contents of the response. The dialogue manager gets input dialogue acts from the NLU and outputs dialogue acts to the NLG. It decomposes the dialogue acts in order to update the information state.

The information state update rules describe grammars for conversational game structure and are written as state charts. We are using State Chart XML (SCXML), a W3C working draft (Barnett et al. 2008), for describing the state charts. SCXML allows for explicit data models that can be manipulated by executable code. This code can be triggered on entry or exit from a state or during a transition. As pointed out by Kronlid and Lager (2007), all these features make it viable to implement the information-state based dialogue model with SCXML.[1]



(a) offer subdialogues

(b) question-answer subdialogues

Figure 5: State charts for *Hassan* domain.

We have defined a set of networks for each type of game/subdialogue. Following Traum and Allen (1994), we model the character's conversational obligations using these networks. Each node indicates the state of the obligations and outgoing arcs with the character as the speaker indicate ways to address these obligations. Figure 5(a) shows a sample network that handles dialogue acts for the offer subdialogue. The outgoing arcs from the currently active states denote all possible dialogue acts that can be generated as a response by the system or can be handled as input from the user. Some of these transitions can be conditional and depend on the data model configuration (i.e. information-state). Although the network structures themselves are currently hand-authored as SCXML documents, some of the constraints for these networks can be authored using the authoring tool as shown in Figure 6.

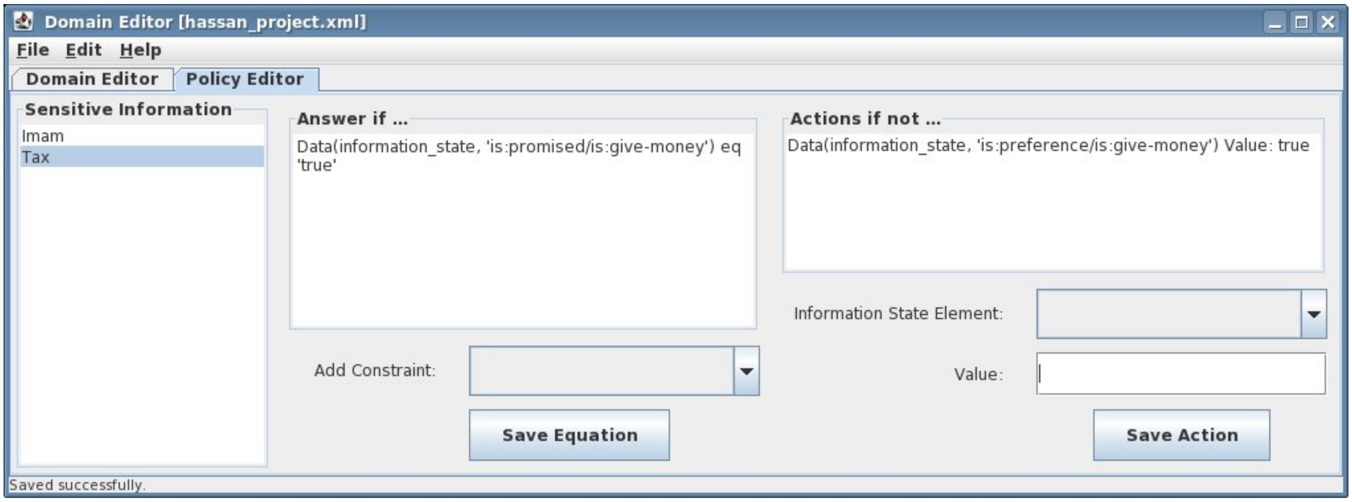
Figure 6 shows the policy editing pane. The leftmost pane lists domain elements that are marked as sensitive information. This can be marked at the level of an object or a specific attribute of an object. For every sensitive information the author can provide a constraint. The constraint can be any boolean expression formed by using the information state elements which can be chosen from a drop-down list. A question about this sensitive information will not be answered till corresponding constraint is satisfied. In case the constraint is not satisfied then actions specified in the rightmost pane are executed. E.g. Any *yn-question* or *wh-question* about the *object* "Tax" will not be answered unless the player has extended the *offer* of "give-money". In case, such a question is asked and the required constraint is not met, then Hassan will set a preference for the offer "give-money", which in turn will result in the next move from Hassan being an *elicit-offer*.

		question resolved, offer not elicited	
1	P	whq	Ok I'm trying to understand where the local taxation is coming from?
		question not resolved, offer not elicited	
2.1	H	grounding	So you want to talk about the taxes.
2.2	H	elicit-offer	I might tell you what you want if there was something in it for me.
		question not resolved, offer elicited	
3	P	offer	We can offer you financial reward.
		question not resolved, offer given	
4.1	H	response-offer	That is very generous of you.
		question not resolved, offer not elicited	
4.2	H	assert	Please understand, I collect taxes for my Imam. All in service to Allah.
		question resolved, offer not elicited	
5	P	whq	And what is his name?
		question not resolved, offer not elicited	
6	H	elicit-offer	My friend, if people find out that I tell you this, it would be a problem for me.

Figure 7: Example dialogue showing the currently active states for the networks in Figure 5. P is the player (human trainee) and H is Hassan.

As an example, in the dialogue from Figure 7, the player asks a sensitive question (utterance 1), the constraints for which are not yet satisfied. At this point as per the authored policy (see Figure 6), Hassan sets the preference for "give-money" offer and chooses to start the offer subdialogue by eliciting that offer (utterance 2.2). After utterance 3 the constraints are met. Hassan can then respond to the offer (*hassan.response-offer* – utterance 4.1) thus completing the offer subdialogue and answer the question (*hassan.assert* – utterance 4.2) thus resolving the question under discussion

[1] We used the apache commons SCXML implementation. [http://commons.apache.org/scxml]



Imam
Tax

Answer if ...: Data(information_state, 'is:promised/is:give-money') eq 'true'

Actions if not ...: Data(information_state, 'is:preference/is:give-money') Value: true

Saved successfully.

Figure 6: Authoring tool can be used to specify the conditions for question-answering network.

and completing the question-answer subdialogue.

We have authored subdialogue networks for greeting, compliment, insult, question-answering, offer, threat, preclosing, closing and grounding subdialogues. Consistent with our design approach of allowing non-experts to rapidly build the dialogue systems, the scenario developer is expected to select from such a set of subdialogues/games for a given domain. A finite set of games can be identified that would cover most of the dialogue phenomena. Still the user is allowed to author subdialogue networks from first principles if needed.

As part of the information state, the dialogue manager maintains which offers or threats have been given. Apart from these each subdialogue maintains appropriate information to conduct that subdialogue. e.g. The question-answer network remembers the last question asked. The dialogue manager also keeps track of the emotional state of Hassan which is composed of emotions like feels-respected, respects-interviewer, social-bonding and fear (Roque and Traum 2007). The transition networks inside the dialogue manager update these emotions based on the incoming dialogue acts. Based on these emotions the character's compliance level is determined as adversarial, reticent or compliant. This compliance level influences what kind of reply will be given. E.g., when adversarial, the character may choose to lie in response to questions, if a lie is available. Apart from emotional state the dialog manager also manages grounding with help of separate set of networks (Roque and Traum 2008).

Textual Level

Natural language understanding and generation converts from surface text to dialogue acts and back again respectively. The authoring tool shown in Figure 2 supports this via links between natural language texts in the bottom pane, and dialogue acts in the middle pane. For each dialogue act from the character, the author can add one or more options for the character to realize this act. Likewise, for the player dialogue acts, the author can link possible ways for the player to produce this act. The Hassan domain has its 102 dialogue acts with Hassan as the speaker connected to 129 surface text utterances. Its 108 dialogue acts with player as speaker are connected to 187 utterances.

The NLU uses a statistical language modeling text classification technique (Leuski and Traum 2008) to map the text produced by the speech recognition to dialogue acts. In case the closest dialogue act match falls below a threshold an *unknown* dialogue act is passed on to dialogue manager. The NLG works in a similar fashion but in reverse direction. Both NLU and NLG require a training corpus of sample utterances linked to dialogue acts, which can be produced using the authoring tool as described above. The task of generating a training corpus for NLU and NLG can be time consuming. It is mitigated by allowing utterances to be linked only to a dialogue act drawn from a specific set of automatically generated dialogue acts. It is easier to choose a dialogue act for an utterance rather than construct one from scratch. As an example consider the dialogue act as shown in Figure 8. Some of these utterances have multiple functions and can be marked up with multiple dialogue acts. But for simplicity, we annotate only the most salient part of the utterance that can be handled by our dialogue manager. Consider utterance 2 in Figure 8, the clause "so that you could do other things that will better benefit allah." does not have any representation in the dialogue act. By avoiding the construction of the dialogue act from scratch and focusing on the most salient part, we can facilitate and speed up the annotation process of such utterances. This produces *consistent* annotations which by design will be handled correctly by the dialogue manager. Some of the utterances shown in Figure 8 are a result of corpus collection through user testing of the virtual human dialogue system. If available, roleplays or WoZ sessions can also be annotated in a similar fashion. If the non-represented parts of these utterances are deemed important, then the domain specification can be expanded to include those using the tool shown in Figure 2. The tool will also automatically generate dialogue acts which will be appropriate elicitations/responses to the new additions, thus ensuring *completeness*.

```
<speech_act speaker="player">
  <primitive_speech_act>
    <offer name="protect-hassan"/>
  </primitive_speech_act>
</speech_act>
```

1 ***I promise you that you will not receive any harm*** *for giving me this information.*

2 *Well I can also help you in other ways and* ***we can protect you*** *so that you could do other things that will better benefit allah.*

3 *Well, if you could help us, the perhaps* ***we could put you in protection. and offer you protective custody*** *because if your people are being taxed unfairly, then you're being taxed unfairly as well too and perhaps we can help.*

4 *Sure I understand, as I said,* ***I can make you safe*** *ah if you're able to share information with me. but ah hopefully that will be enough.*

Figure 8: A sample dialogue act along with the corresponding surface text utterances. The most salient part of these utterances which matches with the dialogue act is highlighted.

Evaluation

Two new characters were built using the authoring tool within a period of a few weeks by subject matter experts who did not have any experience in building dialogue systems. One of these characters is named Amani (see Figure 9), who has witnessed a recent shooting in the marketplace. The trainee is to question her to find out the identity, location and description of the shooter (see Figure 10 for a sample interaction). This Amani domain has 89 dialogue acts with Amani as the speaker and these are connected to 98 utterances which are used in the NLG. The domain also has 113 dialogue acts with player as the speaker linked to 681 utterances which are used in the NLU. We have also built a character named Assad, a local shopkeeper in this marketplace. Since then we have also started to build Mohammed, Amani's brother who will share some domain knowledge with Amani. We expect to use the ability of our tool to re-use the authored content from Amani character. Domain knowledge can be re-used at the object level. All the dialogue acts and the corresponding surface text associated with the object can be re-used. Although some of the surface text may need extra processing for things like indexicals.

Our authoring tool allows to annotate an utterance only with a dialogue act that has been automatically generated from the domain knowledge using a simple dialogue act scheme. To verify the coverage of the scheme, we conducted a dialogue act annotation study for one of our characters, Amani (Artstein et al. 2009). A total of 224 unique players utterances which were collected during system testing were matched by 3 annotators to the closest dialogue act; utterances which did not match an appropriate existing dialogue act were marked with the special *unknown* dialogue act. Overall, 53 of the possible 113 player dialogue acts were selected by at least one annotator as matching at least one player utterance. Inter-annotator agreement was substantially above chance, but fairly low compared to accepted standards: $\alpha = 0.489$ when calculated on individual dialogue acts, and $\alpha = 0.502$ when these were collapsed into dialogue act types indicating illocutionary force alone.[2] However, a detailed analysis of the annotations suggested that some of the disagreements were due to unclear guidelines that do not have an impact on system performance, for example whether a question of the form *Do you know…* or *Can you tell…* should be treated as a *yn-question* or *wh-question*. The analysis also revealed some gaps in the coverage of our dialogue acts scheme, such as the absence of questions which ask about an object without specifying an attribute, as in *Tell me more about the sniper*. Since such questions are very common, constituting nearly 12% of our corpus, we added corresponding dialogue acts to the generation algorithm (Algorithm 1). Overall, the analysis shows that with improved guidelines and extensions, our dialogue act scheme can adequately represent around 80% of actual player utterances. The reader is referred to (Artstein et al. 2009) for further details.

Figure 9: Amani – A virtual human for Tactical Questioning build by using the new authoring tool. The man sitting in the chair is Amani's brother, Mohammed.

Even with the extended dialogue act scheme and improved guidelines, some of the player's utterances will still be marked with the *unknown* dialogue act. Preliminary analysis suggests that it is difficult for annotators to decide whether an utterance can be coerced into one of the existing dialogue acts or whether a new dialogue act needs to be created by extending the domain knowledge of the character. We are currently developing guidelines about when and how to extend the domain to increase the coverage of

[2] Krippendorffs α (Krippendorff 2004) is a generalized measure of interrater agreement, similar to the more familiar K. For a detailed discussion of inter-rater agreement coefficients, see (Artstein and Poesio 2008).

player's utterances. Besides this the dialogue manager has a special network to handle *unknown* dialogue acts which can be caused by out-of-domain utterances or ASR/NLU errors. The dialogue manager attempts to confirm the topic of the conversation and then asks the user to repeat or rephrase. Other strategies to handle *unknown* include taking initiative providing related information about the current topic of conversation if in *compliant* mode or give an *offtopic* response.

Recently, we conducted field testing of Amani at U.S. Military Academy, Westpoint. A total of 33 participants interviewed Amani. These are the users from our target population. In response to the question "In general, Amani responded appropriately to what I was saying." Amani scored 3.24 (mean) on a scale of 1 to 7. For the question "Taken as a whole, Amani was a human-like conversation partner" the score was 3.09. These figures are comparable to third generation Hassan who scored 4.0 and 3.55 respectively (Roque and Traum 2009). Hassan is a character which has been build by several experts over a period of years and through different architectures.

Conclusion

We have described an integrated authoring tool and the accompanying dialogue manger which is used to build several virtual characters for Tactical Questioning. One of the goals is to enable scenario designers to build a dialogue system without the need of expertise in computational linguistics. Our success in building new characters in short amount of time validates the usefulness of the tool and overall architecture.

Acknowledgments

We wish to thank Kevin He and Sarrah Ali for authoring the initial Assad and Amani scenarios, and Michael Rushforth and Jordan Peterson for participating in the annotation study. We also thank two anonymous reviewers for their thoughtful comments and suggestions.

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Army. 2006. Police intelligence operations. Technical Report FM 3-19.50, Department of the Army. Appendix D: Tactical Questioning.

Artstein, R., and Poesio, M. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34(4):555–596.

Artstein, R.; Gandhe, S.; Rushforth, M.; and Traum, D. 2009. Viability of a simple dialogue act scheme for a tactical questioning dialogue system. In *DiaHolmia, 13th Workshop on the Semantics and Pragmatics of Dialogue*.

Barnett, J.; Akolkar, R.; Auburn, R. J.; Bodell, M.; Burnett, D. C.; Carter, J.; McGlashan, S.; Helbing, T. L. M.; Hosn, R.; Raman, T.; and Reifenrath, K. 2008. State Chart XML (SCXML) : State machine notation for control abstraction. http://www.w3.org/TR/scxml/.

Bohus, D., and Rudnicky, A. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *proccedings of Eurospeech-2003*.

Core, M. G., and Allen, J. F. 1997. Coding dialogs with the damsl annotation scheme. In *In Proceedings of AAAI97 Fall Symposium on Communicative Action in Humans and Machines, AAAI*.

Gandhe, S.; DeVault, D.; Roque, A.; Martinovski, B.; Artstein, R.; Leuski, A.; Gerten, J.; and Traum, D. 2008. From domain specification to virtual humans: An integrated approach to authoring tactical questioning characters. In *Interspeech 2008*.

Jordan, P.; Hall, B.; Ringenberg, M.; Cue, Y.; and Rose, C. 2007. Tools for authoring a dialogue agent that participates in learning studies. In *proceedings of AIED 2007*, 43–50.

Krippendorff, K. 2004. *Content Analysis, An Introduction to Its Methodology 2nd Edition*. Sage Publications.

Kronlid, F., and Lager, T. 2007. Implementing the information-state update approach to dialogue management in a slightly extended SCXML. In *Proceedings of the SEMDIAL*.

Larsson, S.; Berman, A.; Hallenborg, J.; and Hjelm, D. 2004. Trindikit 3.1 manual. Technical report, Department of Linguistics, Goteborg University.

Leuski, A., and Traum, D. 2008. A statistical approach for text processing in virtual humans. In *Proccedings of 26th Army Science Conference*.

Leuski, A.; Patel, R.; Traum, D.; and Kennedy, B. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, 18–27.

Lewin, I. 2000. A formal model of conversational game theory. In *Fourth SemDial Workshop: Gotalog 2000*, 115–122.

MITRE, C. 2005. Midiki: MITRE dialogue kit user's manual. Technical report, The MITRE Corporation.

Roque, A., and Traum, D. 2007. A model of compliance and emotion for potentially adversarial dialogue agents. In *The 8th SIGdial Workshop on Discourse and Dialogue*.

Roque, A., and Traum, D. 2008. Degrees of grounding based on evidence of understanding. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, 54–63.

Roque, A., and Traum, D. 2009. Improving a virtual human using a model of degrees of grounding. In *Proceedings of IJCAI-09*.

Sutton, S.; Cole, R.; Villiers, J. D.; Schalkwyk, J.; Vermeulen, P.; Macon, M.; Yan, Y.; Rundle, B.; Shobaki, K.; Hosom, P.; Kain, A.; Wouters, J.; Massaro, D.; and Cohen, M. 1998. Universal speech tools: The CSLU toolkit. In *In Proceedings of ICSLP*.

Traum, D. R., and Allen, J. F. 1994. Discourse obligations in dialogue processing. In *proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*.

Traum, D., and Larsson, S. 2003. The information state approach to dialogue management. In van Kuppevelt, J., and Smith, R., eds., *Current and New Directions in Discourse and Dialogue*. Kluwer.

Traum, D.; Leuksi, A.; Roque, A.; Gandhe, S.; DeVault, D.; Gerten, J.; Robinson, S.; and Martinovski, B. 2008. Natural language dialogue architectures for tactical questioning characters. In *Proceedings of 26th Army Science Conference*.

Xu, W., and Rudnicky, A. I. 2000. Task-based dialog management using an agenda. In *ANLP/NAACL 2000 Workshop on Conversational systems*, 42–47.

			both not greeted, compliment replied, question resolved, offer not elicited, unknown responded, track topic
1	P	greeting	hello
			amani greeted
2	A	greeting	Hello.
			both greeted
3	P	compliment	you have a wonderful house
			compliment given
4	A	response-compliment	This house used to be much nicer before.
			compliment replied
5	P	whq	can i ask you a few questions about the shooting incident yesterday
			question not resolved
6	A	assert	Go on if you can be quick about it.
			question resolved
7	P	unknown	i would like to assure you that you will be protected
			do request repair
8	A	request-repair	You were not clear in what you said. Say it again.
			unknown responded
9	P	offer	i promise to keep you safe
			offer given
10	A	response-offer	You promise me safety.
			offer not elicited
11	P	whq	did you know who did the shooting
			question not resolved, repeat back
12.1	A	repeat-back	So you want to talk about the man, who did the shooting?
			question not resolved, track topic
12.2	A	assert	I have heard that he is called Saif.
			question resolved
13	P	whq	do you know where he lives
14	A	elicit-offer	You are asking for a lot of information. if you expect me to tell you that. You must offer me secrecy.
15	P	offer	i will keep this a secret
16.1	A	response-offer	You have assured me of secrecy.
16.2	A	assert	I believe he hides on the second floor of Assad's shop. I know he is in there. but when i go in the shop i don't see him. And i have no idea where he is now.
17	P	whq	do you know anything about the shooter's daily activities
18.1	A	repeat-back	So you want to talk about the man, who did the shooting?
18.2	A	elicit-offer	As you know, I am a teacher. We are not provided the luxuries of great wealth. One hundred of your dollars would go a long way to help my family make it through these hard times.
			question not resolved, offer elicited
19	P	reject	sorry i'm not authorized to give you any money
			question not resolved, offer not given
20	A	refuse-answer	There is not much i wish to say about that.
			question resolved, offer not elicited

Figure 10: A sample dialogue with Amani. The first line shows currently active states (shown in **bold**) for greeting, compliment, question-answer, offer, unknown and grounding subdialogue networks. In subsequent lines only the change in the active state is shown. P is the player (human trainee) and A is Amani. In Amani domain *Shooter's name, location* and *daily routine* are marked as sensitive information. The *name* will be released only if *safety* is promised. Since *safety* is already offered in utterance 9, it is not elicited after utterance 11. Information regarding *daily routine* will be released only if *give-money* is promised. In utterance 19 player rejects the elicitation of this offer which results in Amani refusing to answer (utterance 20).

Interpreting Two-Utterance Requests in a Spoken Dialogue System

Ingrid Zukerman and **Enes Makalic** and **Michael Niemann**
Faculty of Information Technology
Monash University
Clayton, VICTORIA 3800, Australia
{ingrid,enesm,niemann}@infotech.monash.edu.au

Abstract

This paper describes a probabilistic mechanism for the interpretation of follow-up utterances developed for a spoken dialogue system mounted on an autonomous robotic agent. The mechanism receives as input two utterances and merges them into a single interpretation if possible. For our evaluation, we collected a corpus of hypothetical requests to a robot that includes a large proportion of utterance pairs. The evaluation demonstrates that our mechanism performs well in understanding textual pairs of utterances of different length and level of complexity, but performance is significantly affected by speech recognition errors.

Introduction

The aim of the *DORIS* project (Dialogue Oriented Roaming Interactive System) is to develop a spoken dialogue module for an autonomous robotic agent which assists people in various household tasks. In the future, *DORIS* will engage in full interactions with people and its environment. However, the focus of our current work is on *DORIS*'s language interpretation module called *Scusi?*.

This paper describes the mechanism used by *Scusi?* to interpret pairs of utterances, such as "Get me the mug. It is on the table." or "My mug is on the table. Please get it." The contributions of this paper are:

- A probabilistically grounded process for interpreting a pair of utterances. This mechanism, which can be generalized to multiple utterances, extends our probabilistic process for interpreting a single spoken utterance [Zukerman et al., 2008].
- A formulation for estimating the probability of an interpretation obtained by combining two utterances. This formulation builds on our formalism for estimating the probability of a single utterance, and supports the comparison of merged and un-merged options.

Our evaluation demonstrates that our mechanism performs well in understanding textual pairs of utterances of different length and level of complexity, but performance is significantly affected by speech recognition errors.

This paper is organized as follows. In the next section, we outline the interpretation process for single utterances, and discuss the estimation of the probability of an interpretation. We then extend this probabilistic mechanism for interpreting utterance pairs. The results of our performance evaluation are presented in Section *Evaluation*, followed by a discussion of related research and concluding remarks.

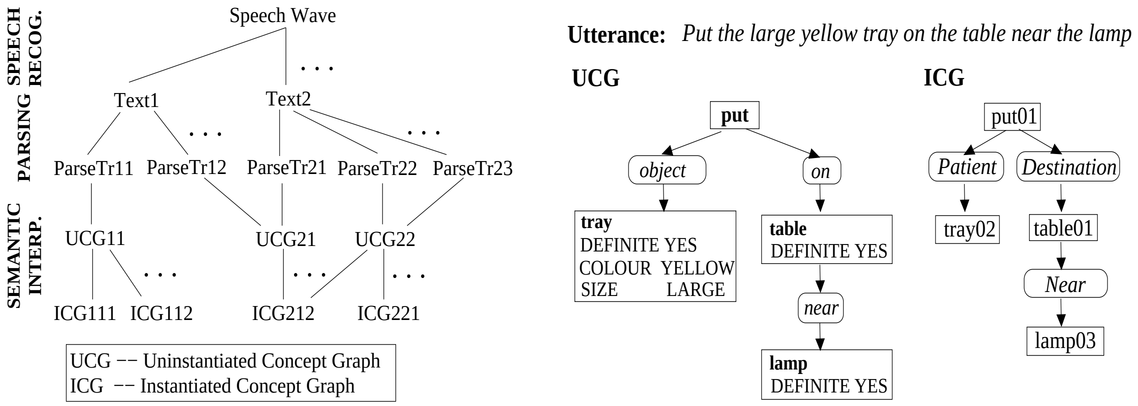
Interpreting a Single Utterance

This section describes the interpretation of a single utterance, thus providing the grounding for the interpretation of utterance pairs.

Scusi? processes spoken input in three stages: speech recognition, parsing and semantic interpretation (Figure 1(a)). In the first stage, it runs Automatic Speech Recognition (ASR) software (Microsoft Speech SDK 5.1) to generate candidate hypotheses (texts) from a speech signal. The ASR produces up to 50 texts for a spoken utterance, where each text is assigned a score that reflects the probability of the words given the speech wave. The second stage iteratively considers the candidate texts in descending order of probability, applying Charniak's probabilistic parser (`ftp://ftp.cs.brown.edu/pub/nlparser/`) to generate parse trees from each text. The parser produces up to 50 parse trees for each text, associating each parse tree with a probability.

During semantic interpretation, parse trees are successively mapped into two representations based on Concept Graphs — a graphical representation whose purpose is "to express meaning in a form that is logically precise, humanly readable, and computationally tractable" [Sowa, 1984].[1] First *Uninstantiated Concept Graphs (UCGs)* are generated, and then *Instantiated Concept Graphs (ICGs)*. UCGs are obtained from parse trees deterministically — one parse tree generates one UCG. A UCG represents syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations are derived from syntactic information in the parse tree and prepositions. Each UCG can generate many ICGs. This is done by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG. Instantiated concepts are objects and actions in the domain (e.g., `mug01, mug02` and `cup01` are

[1] The relationship between Concepts Graphs and predicate logic has been explored in [Dau, 2003].



(a) Stages of the interpretation process

(b) UCG and ICG for a sample utterance

Figure 1: Stages of the interpretation process, and interpretation of a sample utterance

possible instantiations of the uninstantiated concept "mug"), and instantiated relations are similar to semantic role labels [Gildea and Jurafsky, 2002].

Our interpretation process applies a selection-expansion cycle to build a search graph, where each level of the graph corresponds to one of the stages of the interpretation (Figure 1(a)). This process exhibits anytime performance by performing piecemeal expansions: (1) in each selection-expansion cycle, one option is selected (speech wave, textual ASR output, parse tree or UCG), and expanded to the next level of interpretation; and (2) when an option is expanded, a single candidate is returned for this next level, e.g., when we expand a UCG, the ICG-generation module returns the next most probable ICG for this UCG. The selection-expansion process is repeated until all options are fully expanded or a time limit is reached. At any point in this process, *Scusi?* can return a list of ranked interpretations (ICGs) with their parent sub-interpretations (text, parse tree(s) and UCG(s)).

Figure 1(b) illustrates a UCG and an ICG for the request "put the large yellow tray on the table near the lamp". The *intrinsic* features of an object (e.g., colour and size) are stored in the UCG node for this object. *Structural* features, which involve at least two objects (e.g., "the *table* near the *lamp*"), are represented as sub-graphs of the UCG (and then the ICG). This distinction is made because intrinsic features can be compared directly to features of objects in the knowledge base [Makalic et al., 2008], while features that depend on relationships between objects require the identification of these objects and the verification of these relationships. In our example, all the tables and lamps in the room must be considered; the table/lamp combination that best matches the given specification is eventually selected.

Estimating the probability of an ICG

Scusi? ranks candidate ICGs according to their probability of being the intended meaning of a spoken utterance. Given a speech signal W and a context $\mathcal{C}$, the probability of an ICG I, $\Pr(I|W,\mathcal{C})$, is proportional to

$$\sum_{\Lambda} \Pr(T|W)\cdot\Pr(P|T)\cdot\Pr(U|P)\cdot\Pr(I|U,\mathcal{C}) \qquad (1)$$

where T, P and U denote text, parse tree and UCG respectively. The summation is taken over all possible paths $\Lambda = \{T, P, U\}$ from a speech wave to the ICG, because a UCG and an ICG can have more than one ancestor (Figure 1(a)). As mentioned above, the ASR and the parser return an estimate of $\Pr(T|W)$ and $\Pr(P|T)$ respectively; and $\Pr(U|P)=1$, since the process of generating a UCG from a parse tree is deterministic.

The estimation of $\Pr(I|U,\mathcal{C})$ is described in detail in [Zukerman et al., 2008]. Here we present the final equation obtained for $\Pr(I|U,\mathcal{C})$, and outline the ideas involved in its calculation.

$$\Pr(I|U,\mathcal{C}) \approx \prod_{k\in I} \Pr(u|k)\,\Pr(k|k_p,k_{gp})\,\Pr(k|\mathcal{C}) \qquad (2)$$

where u is a node in UCG U, k is the corresponding instantiated node in ICG I, k_p is the parent node of k, and k_{gp} the grandparent node. For example, *Near* is the parent of `lamp03`, and `table01` the grandparent in the ICG in Figure 1(b).

- $\Pr(u|k)$ is the "match probability" between the specifications for node u in UCG U and the intrinsic features of the corresponding node k in ICG I, i.e., the probability that a speaker who intended a particular object k gave the specifications in u (Section *Calculating match probability*).
- $\Pr(k|k_p,k_{gp})$ represents the structural probability of ICG I, where structural information is simplified to node trigrams, e.g., whether `table01` is *Near* `lamp03` (Section *Calculating structural probability*).
- $\Pr(k|\mathcal{C})$ is the probability of a concept in light of the context, which includes information about domain objects, actions and relations.

Scusi? currently handles three intrinsic features: lexical item, colour and size; and two structural features: ownership and several types of locative references. The calculation of the match probability $\Pr(u|k)$ and the structural probability $\Pr(k|k_p, k_{gp})$ is described in detail in [Makalic et al., 2008]. In general, these probabilities are calculated using a distance function (in some suitable space) between the requirements specified by the user and what is found in reality — the closer the distance between the specifications and reality, the higher the probability. Below we outline some of these distance functions, and the probability calculations.

Calculating match probability. For the intrinsic features supported by our system (lexical item, colour and size), the probability of a match between a concept u in UCG U and the corresponding object k in ICG I may be expressed as follows.

$$\Pr(u|k) = \Pr(\mathbf{u}_{\text{colr}}, \mathbf{u}_{\text{size}}, \mathbf{u}_{\text{lex}}|k)$$

The size of an object and its colour depend on its type (e.g., rubbish bins are generally bigger than mugs, and the red in red hair is usually different from the red in a red mug). However, since we are dealing with household objects, we make the simplifying assumption that colour is independent of the type of the object, while size depends on object type. In addition, we weigh the match probabilities for the intrinsic features according to the usage frequency of these features, with more frequent features having a higher weight than less frequent features. This yields the following formulation.

$$\Pr(u|k) = \Pr(\mathbf{u}_{\text{lex}}|k)^{w_{\text{lex}}} \cdot \Pr(\mathbf{u}_{\text{colr}}|k)^{w_{\text{colr}}} \cdot \Pr(\mathbf{u}_{\text{size}}|\mathbf{u}_{\text{lex}}, k)^{w_{\text{size}}}$$

The weights were determined on the basis of the ordering obtained in [Dale and Reiter, 1995], viz *type* $\succ$ *absolute adjectives* $\succ$ *relative adjectives*, where colour is an absolute adjective and size a relative adjective. Specifically, $1 \geq w_{\text{lex}} \geq w_{\text{colr}} \geq w_{\text{size}} \geq 0$.

We employ Leacock and Chodorow's [1998] similarity metric to calculate lexical similarity, and the CIE *Lab* colour space [Puzicha et al., 1999] to estimate colour match. For instance, the (L, a, b) coordinates for `blue`, `azure` and `royal blue` are $(29.6, 68.3, -112.1)$, $(98.8, -5.1, -1.8)$ and $(46.8, 17.8, -66.7)$ respectively, yielding the Euclidean distances *ED*(`blue,royal_blue`)= 70.05, and *ED*(`blue,azure`)=149.5, with the corresponding match probabilities 0.88 and 0.74. Thus, if a blue cup is requested, a royal blue cup has a higher colour match probability than an azure cup. The probability of a size match (e.g., determining whether a particular mug could be described as "large") is estimated by comparing the size of candidate objects with the average size of an object of the requested type [Zukerman, Makalic, and Niemann, 2008].

Calculating structural probability. The overall probability of an ICG structure is decomposed into a product of the probabilities of the trigrams that make up the ICG (the second factor in Equation 2). A trigram consists of a relationship k_p (e.g., ownership, location) between its child concept k and its parent concept k_{gp}, e.g., the trigram `table01`–*Near*–`lamp03` in Figure 1(b) represents the event that `table01` is located near `lamp03`. A structural check then assigns a probability to this event based on the physical coordinates of these objects (the closer the objects, the higher the probability). At present, *Scusi?* considers ownership, e.g., "Sarah's mug" (which can be true, false or unknown), and locative relationships indicated by the prepositions *near, in, on, from, above* and *under*. The mathematical formulation for calculating structural probabilities is detailed in [Makalic et al., 2008].

Algorithm 1 Interpret utterance pairs

```
Require: Speech waves $W_1$, $W_2$, context $\mathcal{C}$
 { Interpret Utterance 1 }
 1: while there is time do
 2:     Generate texts $\{T_1\}$ for $W_1$
 3:     Generate parse trees $\{P_1\}$, and UCGs $\{U_1\}$
 4: end while
 { Interpret Utterance 2 }
 5: while there is time do
 6:     Generate texts $\{T_2\}$ for $W_2$
 7:     Generate parse trees $\{P_2\}$, and UCGs $\{U_2\}$
 8: end while
 9: Determine which utterance is the base and which is the modifier
 { Combine Base and Modifier }
10: $\{U_f^1, U_f^2, \ldots\} \leftarrow$ Merge(base, modifier)
11: if a merger was not possible then
12:     Generate ICGs $\{I_1\}$ and $\{I_2\}$ separately
13:     Select the best interpretations $I_1^*$ and $I_2^*$
14: else
15:     for all merged UCGs $\{U_f^i\}$ do
16:         Generate ICGs $\{I_f^i\}$ on the basis of $U_f^i$
17:     end for
18:     Select the best interpretation $I_f^*$
19: end if
```

Interpreting Utterance Pairs

Algorithm 1 describes the procedure used by *Scusi?* to interpret a pair of utterances. We designate one of these utterances *base* and the other *modifier*. The base utterance generally contains a command (e.g., "Get my mug"), and the modifier provides further information about the base (e.g., "It is in my office."). The base and modifier may appear in any order. At present we assume that the dialogue act (DA) of each utterance has been reliably identified, and focus on the DA `suggest` (aka `request`) for the base utterance, and the DA `clarify` for the modifier. In the future, *DORIS* will apply machine learning techniques to automatically determine the probability of DAs associated with an utterance [Stolcke et al., 2000; Fernández, Ginzburg, and Lappin, 2007].

In Steps 1–8, Algorithm 1 processes each utterance separately up to the UCG stage according to the interpretation process described in Section *Interpreting a Single Utterance*. In order to identify the base and modifier, at present we make the simplifying assumption that one of the utter-

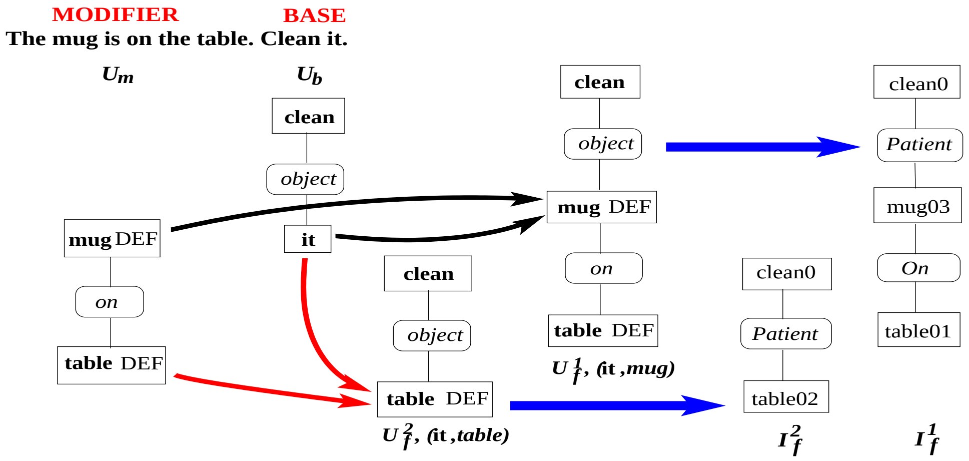


Figure 2: Combining utterances (modifier precedes base)

Algorithm 2 Merge UCGs

```
Require: base and modifier utterances
 1: $i = 1$
 2: while there is time do
 3:     $U_b \leftarrow$ one of the top UCGs for the base
 4:     $U_m \leftarrow$ one of the top UCGs for the modifier
 5:     Create a list of (identifier, referent) pairs for $U_b$ and $U_m$
 6:     for all (identifier, referent) pairs do
 7:         Resolve co-reference between $U_b$ and $U_m$
 8:         $U_f^i \leftarrow$ Incorporate $U_m$ into $U_b$
 9:         $i \leftarrow i + 1$
10:     end for
11: end while
12: return Ranked list of merged UCGs
```

ances in the pair is declarative and the other imperative.[2] This enables Algorithm 1 to use a simple heuristic whereby the imperative utterance is taken to be the *base*, and the declarative utterance the *modifier* (Step 9). This heuristic will be refined in the near future.

Once the types of the input utterances have been determined, Algorithm 2 is activated to merge the UCGs for these utterances (Step 10). If a merger is not possible, ICGs are generated and a top candidate is selected for each utterance separately (Steps 12–13). Otherwise, ICGs are generated from the merged UCGs, and a top candidate is selected (Steps 15–18). In both cases, ICGs are generated by iteratively proposing domain objects and actions for the uninstantiated concepts in a UCG [Zukerman et al., 2008].

The probability of an ICG generated from a single utterance is estimated using Equations 1 and 2. Equation 2 is also employed to estimate the probability of an ICG obtained from a merged UCG. However, in addition to the factors considered in Equation 1, the calculation of the probability of a merged UCG must take into account the probability of the merger (Section *Estimating the probability of a merged interpretation*).

Algorithm 2 returns a list of merged UCGs ranked in descending order of probability. In order to curb combinatorial explosion, only the top few UCGs from the base and modifier utterances are considered, and iteratively paired-up, e.g., first the top (highest probability) base UCG with the top modifier UCG, next the second-top base UCG with the top modifier UCG, and so on (Steps 3–4).

Step 5 of Algorithm 2 generates a list of identifier-referent pairs for a base UCG U_b and its modifier UCG U_m (Section *Resolving references*). The referent is then used to replace the identifier in the appropriate UCG (which could be the base or the modifier). U_b and U_m are merged into a UCG U_f^i by first finding a node n that is common to U_b and U_m, and then copying the subtree of U_m whose root is n into a copy of U_b (Step 8). For instance, given the utterances "The mug is on the table. Clean it." in Figure 2, *Scusi?* produces the list of identifier-referent pairs $\{$(it, *mug*), (it, *table*)$\}$, which yields two intermediate base UCGs `clean`–*object*–`mug` and `clean`–*object*–`table`. Each intermediate base is merged with the modifier UCG using `mug` and `table` as root nodes. This process yields merged UCGs corresponding to "Clean the mug on the table" and "Clean the table" (U_f^1 and U_f^2 respectively in Figure 2, which in turn produce ICGs I_f^1 and I_f^2 among others). If the modifier contains an anaphoric expression, e.g., "Clean the mug on the table. The blue one.", the modifier UCG yields intermediate UCGs, e.g., `[table, COLOUR BLUE]` and `[mug, COLOUR BLUE]`, which are then merged with the base UCG.

Resolving references

Since we are dealing with utterance pairs, we assume that only the second utterance can contain pronouns or one-anaphora, and that these refer to referents in the first ut-

[2] Our procedure can also be applied to pairs of declarative utterances, but we have not considered pairs of imperative utterances — an option that is rare in our dataset (Section *Evaluation*).

terance (references to any other utterance have been disambiguated during manual processing of the data, Section *Evaluation*). In addition, the second utterance may contain noun phrases referring to the first utterance (e.g., "the book"). At present, we handle precise matches of these noun phrases. In the future, we will incorporate Leacock and Chodorow's [1998] scores for approximate lexical matches (Section *Calculating match probability*); such matches occurred in 4% of our test-set.

We use heuristics based on those described in [Lappin and Leass, 1994] to classify pronouns (an example of a non-pronoun usage is "*It* is ModalAdjective that S"), and heuristics based on the results obtained in [Ng et al., 2005] to classify one-anaphora (an example of a high-performing feature pattern is "*one* as head-noun with NN or CD as Part-of-speech and no attached *of* PP"). If a term is classified as a pronoun or one-anaphor, then a list of potential referents is constructed using the head nouns in the first utterance. We use the values in [Lappin and Leass, 1994] to assign a score to each identifier-referent pair according to the grammatical role of the referent in the UCG (this role is obtained from the highest probability parse tree that is a parent of this UCG). For instance, if the referent is the grammatical subject, the score is incremented by 80, and if it is an indirect object, the score is incremented by 40 (all scores start at 100). These scores are then converted to probabilities using a linear mapping function.

Estimating the probability of a merged interpretation

ICGs are ranked according to their probability. Equation 1 estimates the probability of an ICG generated from a single utterance (speech wave) in context $\mathcal{C}$, and Equation 2 estimates the probability of this ICG given a UCG U and the context. To extend these calculations to merged utterances, we must estimate the probability of a UCG obtained by combining two utterances. We then use Equation 2 to estimate the probability of an ICG derived from this UCG.

Let U_f denote the UCG obtained from merging base UCG U_b and modifier U_m using dialogue act D_m of the modifier and identifier-referent pair A. Thus, given speech waves W_b and W_m and context $\mathcal{C}$,

$$\Pr(U_f|W_b, W_m, \mathcal{C}) = \Pr(U_b, U_m, D_m, A|W_b, W_m, \mathcal{C}) \qquad (3)$$

Since a UCG may have more than one ancestor (text or parse tree), Equation 3 may be rewritten as

$$\Pr(U_f|W_b, W_m, \mathcal{C}_m) = \sum_{\Gamma} \Pr(T_b, T_m, P_b, P_m, U_b, U_m, D_m, A|W_b, W_m, \mathcal{C}) \qquad (4)$$

where $\Gamma = \{T_b, T_m, P_b, P_m\}$.

Let us assume that the base precedes the modifier (the converse assumption yields an equivalent formulation to that presented below). We now perform judicious conditionalization, and make the following simplifying assumptions: (1) co-reference resolution depends only on the parse tree of the base and the modifier (according to the rules derived from [Lappin and Leass, 1994; Ng et al., 2005]); (2) given P_m, U_m is independent of U_b; (3) D_m depends on T_m, P_m and the context; and (4) the context affects only the probability of the DA (and later the ICGs, but not in this formula). This yields

$$\begin{aligned}\Pr(U_f|W_b, W_m, \mathcal{C}) = \\ \textstyle\sum_\Gamma \{ &\Pr(T_b|W_b)\cdot\Pr(P_b|T_b)\cdot\Pr(U_b|P_b) \\ &\cdot\Pr(A|P_b, P_m)\cdot\Pr(D_m|P_m, T_m, \mathcal{C}) \\ &\cdot\Pr(T_m|W_m)\cdot\Pr(P_m|T_m)\cdot\Pr(U_m|P_m) \}\end{aligned}$$

where $\Pr(A|P_b, P_m)$ is obtained as described in Section *Resolving references*. At present, we assume that the DA is known, and as mentioned in Section *Interpreting a Single Utterance*, UCGs are deterministically generated from parse trees. This results in the following formulation.

$$\begin{aligned}\Pr(U_f|W_b, W_m, \mathcal{C}) = \\ \textstyle\sum_\Gamma \{ &\Pr(T_b|W_b)\cdot\Pr(P_b|T_b)\cdot\Pr(A|P_b, P_m) \\ &\cdot\Pr(T_m|W_m)\cdot\Pr(P_m|T_m) \}\end{aligned} \qquad (5)$$

Our current mechanism and probabilistic formulation are designed to merge two utterances if possible, and calculate the probability of the resulting interpretation. However, we also need to determine whether a merger is warranted, i.e., whether merging two utterances yields a more promising UCG (and ICGs) than treating them independently. Equations 1 and 5 indicate that to make this determination we just need to compare $\Pr(A|P_b, P_m)$ and $\Pr(D_m|P_m, T_m, \mathcal{C})$ for the merged and un-merged cases (as the probabilities of the tributary UCGs are taken into account in both cases). This requires a probabilistic classification of DAs into: DAs that prescribe different types of mergers (e.g., `correct` and `clarify`), and DAs that indicate no merger. For co-reference resolution, we require a probability for the no-referent case, which is relevant only if the referring expression is a noun phrase (e.g., "the book" could be referring to a previously mentioned book or to a new book).

Evaluation

We first describe our experimental set-up, followed by our results.

Experimental set-up

We conducted a web-based survey to collect a corpus comprising multi-sentence utterances, required for the evaluation of our mechanism. In this survey, we presented participants with a scenario where they are in a meeting room, and they ask a robot to fetch something from their office. The idea is that if people cannot see a scene, their instructions would be more segmented than if they can view the scene. The participants were free to decide which object to fetch and what was in the office. Although their instructions were typed, people employed language that resembled spoken discourse (albeit without disfluencies), because of the way our scenario was presented.

We collected 116 sets of instructions mostly from different participants comprising staff and students at Monash University and the University of Melbourne, friends and family, and acquaintances throughout internet-land (a few people did the survey more than once with different requests). 25 instruction sets comprised only a single utterance, and hence were discarded. Many of the remaining instruction sets had grammatical requirements which exceeded the semantic capabilities of our system (specifically, the capabilities of the procedure that generates UCGs from parse trees). To be able to use these instruction sets, we made systematic manual changes to produce utterance pairs that meet our system's grammatical restrictions (in the future, we will relax these restrictions, as required by a deployable system). Below are the main types of changes we made.

- Sentences with relative clauses were changed to two sentences, e.g., "Get me the book that I left on the desk." was changed to "Get me the book. The book is on the desk."
- Command sequences presented as conjoined verb phrases or sentences were separated into utterance pairs. For example, "I forgot my diary. Go to my desk and pick it up. It is blue." became "(1) I forgot my diary. (2) Go to my desk. (3) Pick it up. (4) It is blue."
- If (as a result of breaking up an instruction set) the first utterance in a pair used an identifier that refers to an earlier item in the instruction set, this identifier was replaced by its referent. For instance, pair 3-4 in the above example produced "Pick up the diary. It is blue."
- Imperative utterance pairs (e.g., pair 2-3 in the above diary example), declarative pairs and pairs that refer to different objects (e.g., pair 1-2 in the diary example) were removed from the test-set (63 pairs were removed in total).
- Composite verbs were simplified, e.g., "I need you to fetch" was changed to "fetch", and "I think I left it on" was changed to "it is on", and out-of-vocabulary composite nouns were replaced by simple nouns or adjectives, e.g., "the diary is A4 size" to "the diary is big".

This process yielded an evaluation test-set of 106 pairs of utterances,[3] where each pair has one declarative and one imperative sentence (Table 1 shows some examples — 'B' stands for base and 'M' for modifier).

M	My important disc is in my office.
B	Please quickly bring me the disc.
B	Please can you fetch my diary in my room?
M	It is on my desk.
B	Can you pick up my notebook?
M	It is on the desk in my office.
M	The cup will be on this desk near the terminal.
B	Pick up the cup.
B	Bring me my diary.
M	The small one.

Table 1: Sample utterance pairs

We then constructed a virtual environment comprising a main office, a small office, a kitchen and a bathroom (Figure 3). Furniture and objects were placed in a manner compatible with what was mentioned in the requests in our corpus; distractors were also placed in the virtual space. In total, our environment contains 183 instantiated concepts (109 objects, 43 actions and 31 relations), including several diaries, books, mugs, phones, keys, pens, jackets, desks, bookcases and computers; and one gun, lamp, statue, bottle, keyboard, monitor and adapter. The (x, y, z) coordinates, colour and dimensions of these objects were stored in a knowledge base.

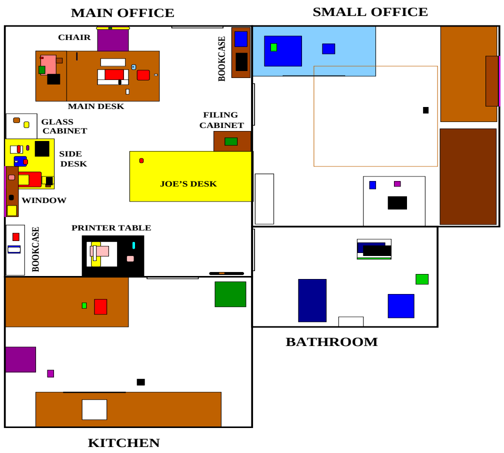


Figure 3: Our virtual environment (top view)

The utterance pairs were then recorded by one of the authors (the ASR software is speaker dependent, and at present we do not handle features of spontaneous speech), and processed by *Scusi?* in the context of our virtual environment. We also processed the text-based utterances separately to determine the effect of ASR error on performance.

Scusi? was set to generate at most 300 sub-interpretations in total (including texts, parse trees, UCGs and ICGs) for each utterance in the test set. A threshold of 85% was applied to the output of the ASR for each utterance, and a threshold of 10% to the output of the parser. These thresholds, which were empirically obtained [Zukerman et al., 2008], prevent the inspection of unpromising options. Specifically, given a threshold *Th*, *Scusi?* stops expanding a node N in the search tree (where N may be the speech wave, a text, a parse tree or a UCG), if the probability of its last-generated child is less than $Th \times \Pr(\text{first child of } N)$. An interpretation was deemed successful if it correctly represented the speaker's intention within the limitations of *Scusi?*'s knowledge base. This intention was represented by one or more *Gold ICGs* that were obtained by manually tagging the ICGs returned by *Scusi?*. Multiple Gold ICGs were allowed if there were several objects that matched a requested item, e.g., "get a mug".

[3] We acknowledge the modest size of this test set compared to that of some publicly available corpora, e.g., ATIS and GeoQuery. However, we must generate our own test set since our task differs in nature from the tasks where these large corpora are used.

Table 2: *Scusi?*'s interpretation performance

Input	**# Gold ICGs with prob. in**		**Average adj.**	**Median adj.**	**75%-ile adj.**	**Not**
	top 1	**top 3**	**rank (rank)**	**rank (rank)**	**rank (rank)**	**found**
Text	80 (75%)	91 (86%)	4.06 (2.17)	0 (0)	1 (0)	1 (1%)
Speech	45 (42%)	53 (50%)	3.73 (1.75)	0 (0)	1 (1)	42 (40%)
Total	106 (100%)	106 (100%)				106 (100%)

Results

Table 2 summarizes our results. Column 1 displays the input type (text or speech). Columns 2-3 show how many utterances had Gold ICGs whose probability was among the top-1 or top-3 probabilities, e.g., the Gold ICG was top ranked in 75% of the cases for textual input. The average *adjusted rank* and *rank* of the Gold ICG appear in Column 4 ("not found" Gold ICGs are excluded from these ranks). The rank of an ICG I is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable ICGs are deemed to have the same position. The adjusted rank of an ICG I is the mean of the positions of all ICGs that have the same probability as I. For example, if we have 3 top-ranked equiprobable ICGs, each has a rank of 0, but an adjusted rank of $\frac{0+2}{2} = 1$. Columns 5 and 6 respectively show the median and the 75%-ile adjusted rank and rank of the Gold ICG, and Column 7 shows how many utterances didn't yield a Gold ICG.

The average adjusted rank of 4.06 for textual input is mainly due to 7 outliers with adjusted rank > 10, with one at adjusted rank 144 (rank 60). All of these outliers and the "not-found" Gold ICG are due to PP-attachment issues, e.g., for the utterance pair "Fetch my phone from my desk. It is near the keyboard.", the top parses and resultant UCGs have "near the keyboard" attached to "the desk" (instead of "the phone"). Nonetheless, the top-ranked interpretation correctly identified the intended object and action in 5 of these 7 cases (when "near" objects are requested, the difference due to PP-attachment is immaterial, as all the items in question are near each other). Median and 75%-ile results confirm that most of the Gold ICGs are top ranked.

Unfortunately, for spoken input only 42% of the Gold ICGs were top-ranked, 50% were ranked top-3, and 40% Gold ICGs were not found. This result may be mainly attributed to the compound ASR error, which was 24% for the first utterance and 36% for the second utterance (regardless of whether it is a base or a modifier). This yields a 0.486 probability of getting a correct top-ranked ASR output for a pair of utterances (indeed, the ASR returned the correct texts in the top rank for 54 of the 106 utterance pairs). Finding 45 top-ranked and 53 top-3 ranked Gold ICGs, which is 83% and 98% of these 54 utterance pairs respectively, means that *Scusi?* was able to pick up additional Gold ICGs relative to its performance for textual input. This indicates that, like the results obtained in [Zukerman et al., 2008] for single utterances, *Scusi?*'s approach of maintaining multiple interpretations overcomes some of the ASR error for utterance pairs. Note that the average ranks (Column 4) obtained for the spoken input are slightly lower (better) than those obtained for the textual input. This is due to the fact that many of the Gold ICGs that had higher (worse) ranks for the textual input were not found for the spoken input. As for textual input, the median and 75%-ile results confirm that when *Scusi?* finds the Gold ICG, its rank is often close to the top.

Related Research

This research extends our mechanism for interpreting stand-alone utterances [Zukerman et al., 2008] to the interpretation of utterance pairs. Our approach may be viewed as an *information state* approach [Larsson and Traum, 2000; Becker et al., 2006], in the sense that utterances may update different informational aspects of other utterances, without requiring a particular "legal" set of DAs. However, unlike these information state approaches, ours is probabilistic. The probabilities returned by our mechanism may be used in conjunction with DA probabilities [Stolcke et al., 2000; Fernández, Ginzburg, and Lappin, 2007] to help a dialogue manager decide whether it is fruitful to merge two utterances, i.e., whether the results obtained from a merger are better than un-merged results (Section *Estimating the probability of a merged interpretation*).

Several researchers in spoken language systems and robot-based systems in particular (e.g., [Matsui et al., 1999; Rayner, Hockey, and James, 2000; Bos, Klein, and Oka, 2003]) take into account expectations from contextual information during language interpretation. Matsui *et al.* [1999] use contextual information to constrain the alternatives considered by the ASR early in the interpretation process. This allows their system to process expected utterances efficiently, but makes it difficult to interpret unexpected utterances. Rayner *et al.* [2000] use contextual information to produce different interpretations from contextually available candidates, and to resolve anaphora and ellipsis. *Scusi?*'s architecture resembles that described in [Rayner, Hockey, and James, 2000] in its successively deeper levels of representation, and its consideration of several options at each level. However, we provide a probabilistic framework for the selection of interpretations, while their selection process is based on diagnostic messages produced during the interpretation process. Finally, Bos *et al.* [2003] developed a dialogue system for a mobile robot called Godot, which understands natural descriptions, and takes context into account. However, unlike *Scusi?*, Godot's language interpretation process uses a logic-based framework, and employs formal proofs for conflict resolution.

Probabilistic approaches to the interpretation of spoken utterances in dialogue systems have been investigated in [Pfleger, Engel, and Alexandersson, 2003; Hüwel and

Wrede, 2006; He and Young, 2003; Gorniak and Roy, 2005] among others. Pfleger *et al.* [2003] and Hüwel and Wrede [2006] employ modality fusion to combine hypotheses from different analyzers (linguistic, visual and gesture), and apply a scoring mechanism to rank the resultant hypotheses. They disambiguate referring expressions by choosing the first object that satisfies a 'differentiation criterion', hence their system does not handle situations where more than one object satisfies this criterion. He and Young [2003] and Gorniak and Roy [2005] use Hidden Markov Models for the ASR stage. However, all these systems employ semantic grammars, while *Scusi?* uses generic, syntactic tools, and incorporates semantic- and domain-related information only in the final stage of the interpretation process. This approach is supported by the findings reported in [Knight et al., 2001] for relatively unconstrained utterances by users unfamiliar with the system, such as those expected by *DORIS*.

Young [Young, 2000] introduced a probabilistic framework for dialogue systems based on Markov Decision Processes (MDPs) and reinforcement learning, focusing on the selection of dialogue acts produced by the system. This approach was extended by Singh *et al.* [2002] and Williams and Young [2007]. In particular, Williams and Young use partially observable MDPs (POMDPs), employing a probabilistic formulation similar to ours (albeit with different conditionalizations). However, these systems focus on slot-filling applications, while *DORIS*'s domain is more open-ended. Nonetheless, the probabilities produced by *Scusi?* could be incorporated into such dialogue systems, as well as into utility-based dialogue systems [Horvitz and Paek, 2000].

Conclusion

We have extended *Scusi?*, our spoken language interpretation system, to interpret pairs of utterances that clarify each others' intent. Specifically, we have proposed a procedure that combines two utterances, and presented a formalism for estimating the probability of the merged interpretation. Our formalism supports the comparison of a merged interpretation with the corresponding un-merged option at the semantic (UCG) and the pragmatic (ICG) stages of the interpretation process. This comparison requires the identification of the DA of an utterance – a task that will be undertaken in the next step of our project. Our mechanism is also well suited for processing replies to clarification questions [Horvitz and Paek, 2000; Bohus and Rudnicky, 2005], as this is a special case of the problem addressed in this paper — the interpretation of spontaneously volunteered, rather than prompted, information.

Although our formalism is probabilistic, only the parse tree probabilities are obtained from frequencies. The other probabilities are obtained from scores that are mapped into the [0,1] range. The use of these scores obviates the need for collecting the large amounts of data required by a wholly frequentist approach.

Our empirical evaluation shows that *Scusi?* performs well for textual input corresponding to (modified) utterance pairs in the target domain, with the Gold ICG(s) receiving one of the top three ranks for most test utterances. However, compound ASR errors have a significant detrimental impact on interpretation performance — an issue we propose to address in the near future. We also intend to expand *Scusi?*'s grammatical capabilities, and implement a procedure for deciding whether utterances should be merged, prior to implementing a full dialogue manager.

References

Becker, T.; Poller, P.; Schehl, J.; Blaylock, N.; Gerstenberger, C.; and Kruijff-Korbayová, I. 2006. The SAMMIE system: Multimodal in-car dialogue. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 57–60.

Bohus, D., and Rudnicky, A. 2005. Constructing accurate beliefs in spoken dialog systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 272–277.

Bos, J.; Klein, E.; and Oka, T. 2003. Meaningful conversation with a mobile robot. In *EACL10 – Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 71–74.

Dale, R., and Reiter, E. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science* 18(2):233–263.

Dau, F. 2003. *The Logic System of Concept Graphs with Negations (And its Relationship to Predicate Logic)*. Lecture Notes in Artificial Intelligence. Heidelberg-Berlin: Springer-Verlag, Volume LNCS 2892.

Fernández, R.; Ginzburg, J.; and Lappin, S. 2007. Classifying non-sentential utterances in dialogue: A machine learning approach. *Computational Linguistics* 33(3):397–427.

Gildea, D., and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28(3):245–288.

Gorniak, P., and Roy, D. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05 – Proceedings of the 7th International Conference on Multimodal Interfaces*, 138–143.

He, Y., and Young, S. 2003. A data-driven spoken language understanding system. In *ASRU'03 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 583–588.

Horvitz, E., and Paek, T. 2000. DeepListener: Harnessing expected utility to guide clarification dialog in spoken language systems. In *ICSLP 2000 – Proceedings of the 6th International Conference on Spoken Language Processing*, 229–229.

Hüwel, S., and Wrede, B. 2006. Spontaneous speech understanding for robust multi-modal human-robot communication. In *Proceedings of the COLING/ACL Main conference poster sessions*, 391–398.

Knight, S.; Gorrell, G.; Rayner, M.; Milward, D.; Koeling, R.; and Lewin, I. 2001. Comparing grammar-based and robust approaches to speech understanding: A case study.

In *EUROSPEECH 2001 – Proceedings of the 7th European Conference on Speech Communication and Technology*, 1779–1782.

Lappin, S., and Leass, H. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics* 20:535–561.

Larsson, S., and Traum, D. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 6:323–340.

Leacock, C., and Chodorow, M. 1998. Combining local context and WordNet similarity for word sense identification. In Fellbaum, C., ed., *WordNet: An Electronic Lexical Database*. MIT Press. 265–285.

Makalic, E.; Zukerman, I.; Niemann, M.; and Schmidt, D. 2008. A probabilistic model for understanding composite spoken descriptions. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, 750–759.

Matsui, T.; Asoh, H.; Fry, J.; Motomura, Y.; Asano, F.; Kurita, T.; Hara, I.; and Otsu, N. 1999. Integrated natural spoken dialogue system of Jijo-2 mobile robot for office services. In *AAAI99 – Proceedings of the 16th National Conference on Artificial Intelligence*, 621–627.

Ng, H.; Zhou, Y.; Dale, R.; and Gardiner, M. 2005. A machine learning approach to identification and resolution of one-anaphora. In *IJCAI-05 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 1105–1110.

Pfleger, N.; Engel, R.; and Alexandersson, J. 2003. Robust multimodal discourse processing. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue*, 107–114.

Puzicha, J.; Buhmann, J.; Rubner, Y.; and Tomasi, C. 1999. Empirical evaluation of dissimilarity measures for color and texture. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, 1165–1172.

Rayner, M.; Hockey, B. A.; and James, F. 2000. A compact architecture for dialogue management based on scripts and meta-outputs. In *Proceedings of the 6th Conference on Applied Natural Language Processing*, 112–118.

Singh, S.; Litman, D.; Kearns, M.; and Walker, M. 2002. Optimizing dialogue management with reinforcement learning: experiments with the NJFun system. *Artificial Intelligence Research* 16:105–133.

Sowa, J. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Stolcke, A.; Coccaro, N.; Bates, R.; Taylor, P.; Ess-Dykema, C. V.; Ries, K.; Shriberg, E.; Jurafsky, D.; Martin, R.; and Meteer, M. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics* 26(3):339–373.

Williams, J., and Young, S. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language* 21(2):393–422.

Young, S. 2000. Probabilistic methods in spoken-dialogue systems. *Philosophical Transactions of the Royal Society A* 358(1769):1389–1402.

Zukerman, I.; Makalic, E.; Niemann, M.; and George, S. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, 581–592.

Zukerman, I.; Makalic, E.; and Niemann, M. 2008. Combining probabilistic reference resolution with simulated vision. In *Proceedings of the ISSNIP 2008 Symposium on Human-Robot Interaction*, 219–224.

Integrating Spoken Dialog with Bayesian Intent Recognition: A Case Study

Ronnie W. Smith
Department of Computer Science
East Carolina University
Greenville, NC 27858, USA
rws@cs.ecu.edu

Brian Adams
Intelligence and Information Systems
Raytheon
Falls Church, VA 22042, USA
brian_adams@raytheon.com

Jon C. Rogers
Integrated Defense Systems
Raytheon
Huntsville, AL 35806, USA
jon_rogers@raytheon.com

Abstract

This paper describes ongoing efforts in developing a spoken natural language dialog system for military planning. The system constructs a detailed representation of the commander's expectations and overall goals for a planned military action. The ultimate goal of the system is to improve a commander's situational awareness through intelligent information filtering based on this representation of commander intent. We focus on the integration of the natural language dialog component with a separate standalone commander intent recognition component that uses a Bayesian network model for intent inference. We present a plan recognition model that correlates utterance level descriptions of basic activities to military objectives that can be translated into evidence usable by the Bayesian intent recognition network. Promising preliminary results on the effectiveness of this integration effort are also provided.

Challenge: Improving Situational Awareness

A never ending challenge for military commanders is to maintain an accurate and timely perception of the critical environmental factors that impact the effectiveness of the battle resources under his command. This perception, termed *situational awareness*, is crucial to effective command during military operations. In recent years, there has been increased investigation into the utility of computational tools to improve situational awareness. This investigation was originally sponsored by DARPA as its Command Post of the Future (CPOF) project.[1]

One avenue of investigation is the use of intelligent information filtering within the context of a set of expectations that a commander has for a planned military action. This set of expectations may be termed the commander's story (Gershon and Page 2001). Having a computational tool that can quickly and accurately classify the overwhelming amount of incoming data and ensure that the commander and his assistants have access to the most important of these data is a much desired goal. One essential element of this tool is the ability for the computer system to automatically capture the commander's story as the planning process ensues prior to plan execution. While the ability is also needed to capture the plan modification process based on actual events, that is beyond the scope of the current work.

This paper reports on some of the technical challenges of building a spoken natural language dialog system for military story capture, the CPOF Story Capture System (CPOF-SCS). We first provide some background on natural language interfaces for military command and control, including an earlier version of CPOF-SCS that only focused on low-level military activities. The paper then describes the construction of a standalone Bayesian intent network for inferring higher-level commander intent and construction of the required components for enhancing CPOF-SCS to be able to produce evidence for this intent network.

Background: Natural Language Interfaces for Military Command and Control

Early efforts at providing natural language interfaces to military planning applications focused on the design of military simulations of possible actions. QuickSet (Johnston et al. 1997) is a multimodal interface that uses a unification based approach over typed feature structures for determining the appropriate interpretation. Due to the command-driven nature of the application, a great deal of functionality can be achieved without a complex model of the ongoing dialog.

CommandTalk (Stent et al. 1999) is primarily a spoken natural language dialog system for the same application. Its gestural capabilities are limited to the specification of points on a map. The dialog manager is a collection of finite state machines (FSM) that coordinate initiative and handle parameter specification. Neither system is designed to be able to maintain a representation of an actual planned military action for the purpose of helping commanders maintain appropriate situational awareness via the use of intelligent information filtering and reporting.

Background: Dialog for Story Capture of Activities

As reported in (Smith et al. 2002) the initial CPOF-SCS prototype focused on the acquisition and representation of a military *activity*. Activities are specific actions a military

[1] Our work was originally supported in part by DARPA contract F30602-99-C-0060 (through subcontract from General Dynamics: Advanced Information Systems). Work has continued on this system as it serves as an interesting testbed for ideas.

force has been ordered to carry out. An example would be moving a specific force to a location. Activities are tied to explicit orders from a commander that may or may not directly provide information about commander intent. Examples of activities include movement, position establishment, and reconnaissance. Associated with an activity type are parameters, some of which are mandatory and some are optional. The dialog system will attempt to continue interaction about the current activity until values for all mandatory parameters are supplied. This approach is an instantiation of the Missing Axiom Theory of dialog (Smith, Hipp, and Biermann 1995).

These activities act as expectations for the commander as to how the overall military action will unfold. As noted before, we can think about these expectations as a story (Gershon and Page 2001). Consequently, a possible machine representation for these expectations is a *story graph*, where the nodes of the graph represent expectations at particular moments in time and the arcs represent change of expectations between nodes. An example set of expectations in a node might be the following.

```
1. Delta company assumes an attack by
   fire position in building B.
2. Alpha company assumes an attack by
   fire position on hill C.
3. Echo company assumes an observation
   position on hilltop D.
```

Through its integration with knowledge bases for the geographic and military entities relevant to a particular situation, the system is able to process utterances that are voice only, voice and gesture, and gesture only. While the prototype CPOF-SCS can capture and represent information about activities, it cannot represent information about higher-level commander intent—the higher level strategic reasons for the specific activities. For example, a series of movements and position establishments may have the purpose of cutting off the lines of retreat for the enemy. This strategic goal would represent the ultimate purpose of all the activities, even if it was not explicitly stated. In order to be part of an effective tool for intelligent information filtering, capture and representation of commander intent is also necessary. The approach we have taken for addressing this problem is to first develop a Bayesian network model for inferring commander intent. Initially, this model relied on hand generated evidence, but ultimately, this evidence must be generated by CPOF-SCS automatically. We next report on the enhancements to CPOF-SCS that enable this to happen.

Bayesian Networks for Commander Intent Recognition

We adopt the following definition of commander intent as provided in a military field manual (FM 101-5-1 1997).

> Commander's personal expression of why an operation is being conducted and what he hopes to achieve. It is a clear and concise statement of a mission's overall purpose, acceptable risk, and resulting end state (with respect to the relationship of the force, the enemy, and the terrain). It must be understood two echelons below the issuing commander because it provides an overall framework within which subordinate commanders may operate when a plan or concept of operation no longer applies, or circumstances require subordinates to make decisions that support the ultimate goal of the force.

To paraphrase the definition provided above, *commander intent* is the ultimate goal of an operation or the plan when the plan no longer applies. Commander intent can be thought of as the highest level or ultimate plan during a military operation (Adams 2006).

In order to begin to develop a computational model for recognizing commander intent, it was necessary to collect data from experienced military commanders. This data collection was conducted at the School for Advanced Military Studies at Fort Leavenworth, Kansas. The data collection instrument was a set of three Tactical Decision Games (TDGs) from the *Marine Corps Gazette*. A TDG is a tool used by the military to train officers and consists of a written scenario description, a map, and a time constraint. The scenario provides information such as your identity, your goal, your available resources, and relevant enemy information. An officer is expected to produce a solution containing a set of orders and objectives within the time allotted to accomplish the goal. A TDG is a critical thinking and reasoning exercise. Consequently, a TDG is a useful tool for studying commander intent.

Because of the inherent uncertainty in the recognition of commander intent,[2] our commander intent recognition models are based on Bayesian networks. Bayesian networks are a form of belief networks. The network implementation is a Directed Acyclic Graph (DAG) where the nodes are classified as *intent nodes* or *information nodes*. Intent nodes correspond to the possible intents that are relevant to the battle scenario. The information nodes are the set of observable objectives in the battle scenario. The "arcs can be thought of as causal or inference links" (Albrecht, Zukerman, and Nicholson 1998). In our case the inference links connect information nodes to intent nodes, and in some cases intent nodes to other intent nodes.

Figure 1 shows a simplified version of one of the networks developed. The network is based on a TDG with the same name (Graves 2001). The partial network is shown with intent nodes displayed as rectangles and information nodes displayed as ovals. A description of each node is provided in Figure 2. With respect to terminology, SBF means Support By Fire, ABF means Attack By Fire, and Obj means Objective. Each TDG scenario requires its own intent recognition network.

An example inference about commander intent would be that evidence for the objective 2ndCrossRiver implies that possible commander intents include IsolateEast and SecureBridge. Furthermore, following the arcs from these two intent nodes implies that other possible commander intents include IsolateWest and PreventMovement.

[2] This uncertainty was confirmed during our debriefing interviews with our military commander subjects from the previously mentioned data collection activity.

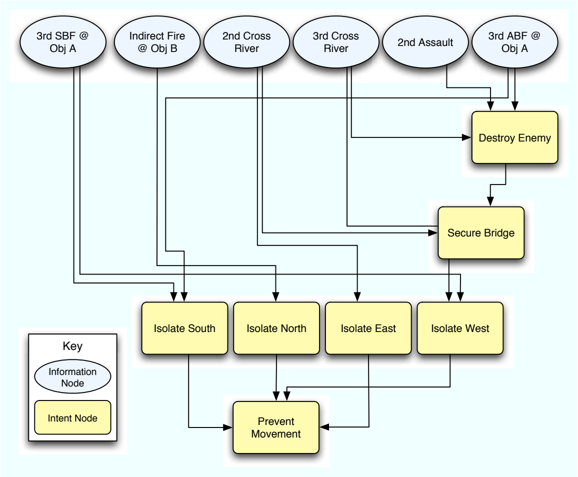


Figure 1: Simplified Commander's Intent Network

The Commander's Intent Network (Simplified)
13 total nodes (6 information, 7 intent)
Information Nodes

3rdSBF@ObjA Third platoon establish a SBF position oriented at Obj. A

3rdABF@ObjA (Third platoon establish an ABF position oriented at Obj. A)

2ndAssault (Second platoon leads an assault against enemy forces)

2ndCrossRiver (Second platoon crosses the Minse River)

3rdCrossRiver (Third platoon crosses the Minse River)

IndirectFire@ObjB (Indirect fire aimed at Obj. B or north or Obj. B)

Intent Nodes

IsolateEast (Isolate Obj. A to the east)

SecureBridge (Secure the bridge crossing)

IsolateSouth (Isolate Obj. A to the south)

IsolateWest (Isolate Obj. A to the west)

DestroyEnemy (Destroy the main enemy presence in the vicinity of Obj. A)

IsolateNorth (Isolate Obj. A to the north)

PreventMovement (Prevent enemy forces from manuevering)

Figure 2: Commander's Intent Network Description

After the data collection was completed, the model was developed, a software prototype implemented, and results were gathered that show that the model supports the ability to efficiently and correctly infer commander intent (Rogers 2003). However, the model relied on manual construction of the evidence to be transmitted to the network. We next describe the enhancements made to CPOF-SCS to automatically produce evidence based on the ongoing dialog.

Augmenting CPOF-SCS with Intent Recognition

Information Node Evidence as Military Objectives

In order to successfully recognize commander intent, CPOF-SCS must be able to take the utterance and dialog semantics of utterances, and generate evidence for the information nodes of the intent recognition network. The key idea in being able to translate information about activities into evidence for information nodes is to use the information about activities to recognize *military objectives*. We use the following definition of a military objective.

> 1. The physical object of the action taken (for example, a definite terrain feature, the seizure or holding of which is essential to the commander's plan, or, the destruction of an enemy force without regard to terrain features). 2. The clearly defined, decisive, and attainable aims which every military operation should be directed towards. (FM 101-5-1 1997)

Based on the analysis of the data collected from TDG exercises, there were three main categories of objectives: (1) Establish Position; (2) Offensive Action; and (3) Movement. By no means do we claim these are the only possible military objectives—these are simply the main ones we observed during our data collection efforts and are thus our initial focus. These categories generalize over the entire set of exercises, and consequently are usable for a broad variety of military plans. The classification of each information node in the network of figure 1 is presented in table 1.

Information Node	Activity	Objective Category
3rdSBG@ObjA	Support by Fire Position	Establish Position
3rdABF@ObjA	Attack by Fire Position	Establish Position
2ndAssault	Assault	Offensive Action
2ndCrossRiver	Movement	Movement
3rdCrossRiver	Movement	Movement
IndirectFire@ObjB	Indirect Fire Attack	Offensive Action

Table 1: Objective Classification for Intent Nodes: Commander's Intent Network

In order to bridge the gap between the very general set of possibilities of multimodal spoken inputs with the scenario-specific possible commander intents, we have enhanced CPOF-SCS with a plan recognition process that can be used to recognize objectives from fully specified activities acquired during interaction with the system. These objectives can then be translated into evidence for the Bayesian intent network. After presenting information on the plan types and structure, we will describe an architectural enhancement to CPOF-SCS, the Resolution System, for handling the myriad of knowledge sources for force and geographic information that may be needed for a general military planning system. Finally, we will describe the operation and performance of the enhanced CPOF-SCS for automatically performing intent recognition.

Plan Types

Plans represent a series of actions that attempt to achieve a goal. In this domain, the goal is the military objective that corresponds to the plan. For example, two activities, moving to a position, and attacking an enemy force represents a military objective to establish an attack by fire position at the specified position.

The above example is an example of a *conjunctive plan* since it requires the specification of multiple activities to infer the objective. A *direct plan* is one where a single activity directly specifies the objective. For example, a specified activity of moving a force to a location would correspond directly to a movement objective. Finally, an *indirect plan* is one where some form of indirect relationship exists between the activity and an objective. There are two main forms of indirect plan. In the first case, the activity contains an indirect representation of an alternative entity. For example, consider the case where a building X contains force Y and is also part of a larger objective C. A stated activity such as "Alpha move to building X" would directly imply an objective of a movement to not only building X, but also to objective C. In the second case, one activity can be inferred from another activity. For example, establishing an attack by fire position (an Establish Position objective) implies attacking the target (an Offensive Action objective). For the situation given above, consider an alternative activity statement, "Alpha establish an attack by fire position at objective C." In this case, if an objective is recognized corresponding to Alpha establishing an attack by fire position oriented at Objective C, another recognized objective should be that Alpha is engaging in an Offensive Action targeting Objective C. There is also an example of the first type of indirection as the attack on Objective C also implies an attack on force Y who is located inside objective C.

Plan Structure

The model we have adopted is similar in structure to the one described by (Blaylock 2001). In our model, plans have three main components. The components are: a set of activities, a military objective solution, and a set of constraints. Each of the three components fulfills a critical function in the plan recognition process. The activities model the inputs to the plan and drive the recognition process. The military objective component models the plan that will be transmitted to the evidence generator so that it can be related to the intent network. The constraints validate the plan. Examples of possible constraints are the following:

- That the activity is direct.
- That a location is specified
- That a destination location is specified (in the case of a movement activity).
- That a force is specified.
- That the force completing the activity is a friendly force (as opposed to an enemy force).
- That conjoined activities occur in the correct order.
- That the same force is completing both activities (e.g. both moving and attacking).

Context Usage for Name Resolution

Consider the utterance "Move Alpha to Checkpoint Alpha One starting one hour after that." The Dialog Controller module of CPOF-SCS must consult with multiple knowledge sources in order to resolve the meaning of names (e.g. "Checkpoint Alpha One") and times (e.g. the event referent associated with "that" in "starting one hour after that"). Furthermore, while the CPOF-SCS prototype including all knowledge repositories was built by our research team, the ultimate system design must plan for utilizing knowledge repositories possibly provided by other system developers. Consequently, CPOF-SCS was enhanced with a Resolution System module that processes requests for contextual information from all knowledge sources. In this fashion a common interface can be provided that allows referents to be resolved either internally or externally. An external source is defined as one that requires information to be passed over a communications link and is external to the Dialog Controller and plan recognition process. An internal source is one that is tightly coupled to the Dialog Controller and can be accessed through local storage structures through language constructs such as method invocation.

For the purposes of discussion, let us focus on two potential inputs: geographic entities and military forces. For geographic entities, the context is static and scenario-dependent. Thus it is most likely that the information is already present in an external data source. Whenever a reference to a geographic entity is made, the system must ask an external entity for the complete representation of the entity. When a complete representation is received, the local representation can be expanded to include all information stored in the repository representation.

Certain information such as military forces may be stored internally by the Dialog Controller. The key is that military forces move over the course of the scenario. Generally, geographic entities do not move over the course of a scenario. Alternatively, military forces move frequently. The context for the location of a force is dynamically determined during the ongoing dialog. There is a distinction between theoretical movement of forces as specified in a plan and the actual location of forces at the current moment in time prior to the execution of the plan (information for which is likely in an external database). Thus, context information comes from internal mechanisms as well as external repositories.

To further motivate the necessity for a single interface, consider references with alternative representations. Because multiple geographic entities and military forces can occupy a common location, a reference to one may indirectly reference another. Further, there may be a situation where a name can be used to represent two entities. Consider the utterance, “Have Bravo establish a support by fire position oriented at Objective A.” It is ultimately the enemy inside of Objective A and not the ground in the area of Objective A that requires fire. Knowing the relationship of one entity to others requires additional information that is stored in potentially two sources. By having a single interface, the plan recognition model needs only to deal with what context information to resolve and not where to resolve it.

The system implemented for our computational model, the Resolution System, facilitates the issuing and processing of the information requests used to obtain additional information. Essentially, the Resolution System deals with interfacing with multiple internal repositories and external repositories transparently. Further, the Resolution System correlates retrieved information with the requesting reference. Thus, the plan recognition model implementation is free of context request and application functionality and can deal specifically with plan recognition.

Plan Recognition Process

We will illustrate the process through the processing of the following utterance sequence that relates to the Commander’s Intent Network of Figure 1.[3]

```
U1: Have 3rd platoon move across the
    river to here.
    (gesture via mouse click)
U2: Have them attack Objective A.
```

First, the basic utterance level processing mechanisms for language, gesture, and their merger transmits the utterance-level semantics to the Dialog Controller. Besides using internal reference resolution mechanisms to disambiguate the word “them” from the second utterance, the Dialog Controller will also use the Resolution System to handle the resolution of names (e.g., 3rd Platoon and Objective A) before the plan recognition process begins. Plan recognition takes place on an utterance by utterance basis.

The key step in the process is the matching of a completed activity acquired through user input with the specification for each relevant plan framework. Each framework corresponds to a different possible objective. In this case, there is an empty plan representing the framework for establishing an attack by fire position. The plan involves moving to a location and attacking an enemy from the new location. When U1 is received, the plan is partially completed as the first activity is satisfied. When U2 is received, all activities are satisfied and the plan is complete. Thus the plan corresponding to 3rd platoon establishing an attack by fire position is recognized. In addition, note that U1 by itself leads to recognition of a plan for a movement objective. During the processing of any given utterance, the recognition of multiple plans may be ongoing. Furthermore, as each new activity is processed, it must be matched against not only empty plan frameworks, but also partially filled plan frameworks (such as the attack by fire framework that was partially filled after utterance U1, and completed by the activity described in utterance U2). Although in this example, each utterance specifies a complete activity, it can require several utterances to specify an activity that can be used by the plan recognition process.

All plan frameworks that are updated during the processing of a newly acquired activity are part of the set of plans known as the *evaluation set* that are used in the plan completion evaluation algorithm described in Figure 3. As shown, the focus of the algorithm is finding valid and completed plans and producing the external representation for these plans. The representation is used by the external evidence generation component to generate inputs to the Bayesian intent recognition network. This evidence is transmitted in XML format. A matching process between the evidence and the network is used to determine if the new evidence is relevant to any of the information nodes in the network.

Results

The first step in validating the results was to test the performance of CPOF-SCS on the same tests used for the system that relied on manually constructed evidence reported in (Rogers 2003). In terms of the traditional measures of

[3] To simplify the presentation, issues related to time will be omitted.

```
1. For each Plan in the Evaluation Set.
  (a) For each Activity in the plan:
    i. If the activity is uninstantiated;
       Move to next plan (Go to step 1).
  (b) For each Constraint in the plan:
    i. If the constraint is violated;
       Delete the plan.  (Go to step 1).
  (c) Produce an Objective representation
      and transmit it to the Evidence
      Generation Module.
  (d) Mark the plan as completed
```

Figure 3: Plan Completion Evaluation

recall and precision relative to the expected and possible intents recognized, CPOF-SCS performed at least as well as the original system.

To further confirm the performance of CPOF-SCS and to assess the generality of the model, a completely different scenario was evaluated. This scenario was based on a demonstration provided by General Holcomb (Holcomb 2000). This scenario is based on an urban military action from the Vietnam War. An intent network and test interaction was independently developed by a research team member not involved in the development of the intent recognition component of CPOF-SCS. The results of this test are reported in detail in (Adams 2006). Recall and precision values in excess of 85% for the twenty relevant commander intents were achieved in all cases.

Future Work

While promising, much work remains. Although CPOF-SCS can now recognize commander intent, it does not have the ability to provide usable interactive feedback to a commander as the planning dialog ensues. Work on enhancing the dialog capabilities to provide such feedback, and to handle the ensuing clarifications and corrections is needed. While some work on developing a tool for producing general intent networks has been done (Graham 2006), more work is needed. In addition, usability testing with a broader set of participants is also required. Finally, incorporation into a larger software tool that can take the system's story graph representation and perform intelligent information filtering is also required. At present, the intelligent information filtering is restricted to scripted tests.

Acknowledgements

We would like to express our appreciation to other project team members from East Carolina University, General Dynamics, and Duke University who have collaborated with us on the idea of military stories and information filtering. In particular, we note the major contributions of Brian Manning, Steve Graham, and Niels Kasch of East Carolina, Amaury Alvarez of General Dynamics, and Dr. Alan Biermann of Duke University.

We would also like to acknowledge the constructive comments of the anonymous reviewers of this paper. Their suggestions have been very helpful in producing the final version of this paper.

References

Adams, B. 2006. Activity based multimodal plan recognition of commander intent. Master's thesis, East Carolina University.

Albrecht, D.; Zukerman, I.; and Nicholson, A. 1998. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction* 5–47.

Blaylock, N. 2001. Retroactive recognition of interleaved plans for natural language dialogue. Technical Report 761, University of Rochester.

U.S. Marine Corps. 1997. *FM 101-5-1/MCRP 5-2A: Operational Terms and Graphics*. Available online at http://www.fas.org/man/dod-101/army/docs/fm101-5-1/f545con.htm.

Gershon, N., and Page, W. 2001. What storytelling can do for information visualization. *Communications of the ACM* 31–37.

Graham, S. 2006. Automating Bayesian intent network generation. M.S. Project Report: East Carolina University.

Graves, T. 2001. Tactical decision game #01-7: The commander's intent. *Marine Corps Gazette* 85(7):86.

Holcomb, G. 2000. TDG at IDA. Videotaped Demonstration.

Johnston, M.; Cohen, P.; McGee, D.; Oviatt, S.; Pittman, J.; and Smith, I. 1997. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 281–288.

Rogers, J. C. 2003. Developing Bayesian networks for recognition of commander intent. Master's thesis, East Carolina University.

Smith, R.; Manning, B.; Rogers, J.; Adams, B.; Abdul, M.; and Alvarez, A. 2002. A dialog architecture for military story capture. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, 184–187.

Smith, R.; Hipp, D.; and Biermann, A. 1995. An architecture for voice dialog systems based on Prolog-style theorem-proving. *Computational Linguistics* 21:281–320.

Stent, A.; Dowding, J.; Gawron, J. M.; Bratt, E.; and Moore, R. 1999. The CommandTalk spoken dialogue system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 183–190.

Open-World Dialog: Challenges, Directions, and Prototype

Dan Bohus and Eric Horvitz

Microsoft Research
One Microsoft Way
Redmond, WA, 98052

Abstract

We present an investigation of *open-world dialog*, centering on building and studying systems that can engage in conversation in an open-world context, where multiple people with different needs, goals, and long-term plans may enter, interact, and leave an environment. We outline and discuss a set of challenges and core competencies required for supporting the kind of fluid multiparty interaction that people expect when conversing and collaborating with other people. Then, we focus as a concrete example on the challenges faced by receptionists who field requests at the entries to corporate buildings. We review the subtleties and difficulties of creating an automated receptionist that can work with people on solving their needs with the ease and etiquette expected from a human receptionist, and we discuss details of the construction and operation of a working prototype.

1. Introduction

Most spoken dialog research to date can be characterized as the study and support of interactions between a single human and a computing system within a constrained, predefined communication context. Efforts in this space have led to the development and wide-scale deployment of telephony based, and more recently multimodal mobile applications. At the same time, numerous and important challenges in the realm of situated and open-world communication remain to be addressed.

In this paper, we review challenges of dialog in *open-world* contexts, where multiple people with different and varying intentions enter and leave, and communicate and coordinate with each other and with interactive systems. We highlight the opportunity to develop principles and methods for addressing these challenges and for enabling systems capable of supporting natural and fluid interaction with multiple parties in open worlds—behaviors and competencies that people simply assume as given in human-human interaction. We begin by reviewing the core challenges of moving from *closed-world* to *open-world* dialog systems, and outline a set of competencies required for engaging in natural language interaction in open, dynamic, relatively unconstrained environments. We ground this discussion with the review of a real-world trace of human-human interaction. Then, we present details of a prototype open-world conversational system that harnesses multiple component technologies, including speech recognition, machine vision, conversational scene analysis, and probabilistic models of human behaviour. The system can engage in interaction with one or more participants in a natural manner to perform tasks that are typically handled by receptionists at the front desk of buildings. We describe the set of models and inferences used in the current system and we highlight, via review of a sample interaction, how these components are brought together to create fluid, mixed-initiative, multiparty dialogs.

2. Open-World Dialog

To illustrate several challenges faced by open-world dialog systems, we shall first explore real-world human-human interactions between a front-desk receptionist and several people who have arrived in need of assistance. We focus on a representative interaction that was collected as part of an observational study at one of the reception desks at our organization. The interacting parties and physical configuration are displayed in the video frame in Figure 1.

At the beginning of the segment, the receptionist is on the phone, handling a request about scheduling a conference room, viewing availabilities of rooms and times on her computer in support of the request. Participant 1 (P_1) is an external visitor who the receptionist has just finished speaking with; he is currently filling in a visitor registration form. As P_1 is completing the form, the receptionist answers the telephone and engages in a phone conversation with participant 4 (P_4). During this time, participant 2 (P_2) enters the lobby from inside the building, approaches the reception desk, and makes eye contact with the receptionist. The receptionist, knowing that P_1 needs additional time to complete the registration form, and that the conversation can continue with P_4 while she engages in a fast-paced interaction with P_2, moves to engage with P_2. Apparently relying on inferences from the observation that P_2 came from inside the building, the receptionist guesses that P_2 most likely needs a shuttle to another building on the corporate campus. She lifts her gaze towards P_2 and asks P_2 softly (while moving her mouth away from the phone microphone), "Shuttle?" P_2 responds with a building number.

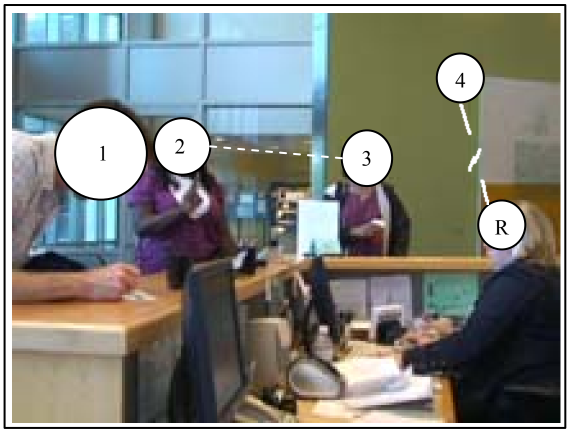


Figure 1. Video frame from a multiparty interaction.

While the receptionist continues on the phone with P_4 on options for arranging a meeting room in the building, she interacts with a shuttle ordering application on the computer. Soon, participant 3 (P_3) approaches the reception desk. At this time, P_2 re-establishes eye contact with the receptionist and indicates with a quick hand gesture and a whisper that the shuttle is for two people. The receptionist now infers that P_2 and P_3—who have not yet displayed obvious signs of their intention to travel together—are actually together. The receptionist whispers the shuttle identification number to P_2 and continues her conversation with P_4, without ever directly addressing P_3. Later, once P_1 completes the form, the receptionist re-engages him in conversation to finalize his badge and contact his host within the building.

The interaction described above highlights two aspects of open-world dialog that capture key departures from the assumptions typically made in traditional dialog systems. The first one is the *dynamic, multiparty* nature of the interaction, *i.e.*, the world typically contains not just one, but multiple agents who are relevant to a computational system, each with their own goals and needs. The second departure from traditional dialog systems is that the interaction is *situated*, *i.e.,* that the surrounding physical environment, including the trajectories and configuration of people, provides rich, relevant, streaming context for the interaction. Our long-term goal is to construct computational models that can provide the core skills needed for handling such situated interaction in dynamic multiparty settings, and work with people with the etiquette, fluidity and social awareness expected in human-human interactions.

In the following two subsections, we discuss the multiparty and situated aspects of open-world interaction in more detail, and we identify the challenges and opportunities that they frame. In Section 3, we review these challenges and outline a set of core competencies required for open-world dialog. Then, in Sections 4 and 5, we describe a prototype situated conversational agent that implements multiple components of an open-world dialog and review their operation in the receptionist setting.

2.1. Multiparty Aspect of Open-World Dialog

The assumption in spoken dialog research to date that only one user interacts with the system is natural for telephony-based spoken dialog systems and is reasonable for a large class of multimodal interfaces. In contrast, if we are interested in developing systems that can embed their input and interaction into the natural flow of daily tasks and activities, the one-user assumption can no longer be maintained.

The open world typically contains more than one relevant agent. Each agent may have distinct actions, goals, intentions, and needs, and these may vary in time. Furthermore, the open world is dynamic and asynchronous, *i.e.,* agents may enter or leave the observable world at any point in time, and relevant events can happen asynchronously with respect to current interactions.

The flow of considerations from single-user, closed-world systems to increasingly open worlds is highlighted graphically in Figure 2. Systems providing service in the open world will often have to have competencies for working with multiple people, some of whom may in turn be coordinating with others within and outside an agent's frame of reference. Such a competency requires the abilities to sense and track people over time, and to reason jointly about their goals, needs, and attention. We can categorize interactive systems based on the assumptions they make regarding the number and dynamics of relevant agents and parties involved in the interaction as follows:

- *Single-user interactive systems* engage in interaction with only one user at a time. Traditional telephony spoken dialog systems, as well as most multimodal interfaces such as multimodal mobile systems, *e.g.* [1, 26], multi-modal kiosks *e.g.* [9, 13], or embodied conversational agents *e.g.* [5] fall into this category.
- *Fixed multi-participant interactive systems* can interact with one or more participants at a given time. The number of participants in a given interaction is known in advance.
- *Open multi-participant interactive systems* can interact with one or more participants. Participants may leave or join an interaction at any given time.
- *Open multiparty interactive systems* further extend the class of open multi-participant systems in that they can engage in, pursue, and interleave multiple parallel interactions with several different parties. The receptionist interaction discussed earlier falls into this last category, as does the prototype system we shall discuss later, in Sections 4 and 5.

The pursuit of multi-participant and multiparty interactive systems brings to fore several research challenges. First, the multi-participant aspect adds a new dimension to several core dialog system problems like dialog management, turn taking, and language understanding. Current solutions for these problems typically rely on the single-user assumption and do not generalize easily to the multi-participant case. We also face entirely new types of prob-

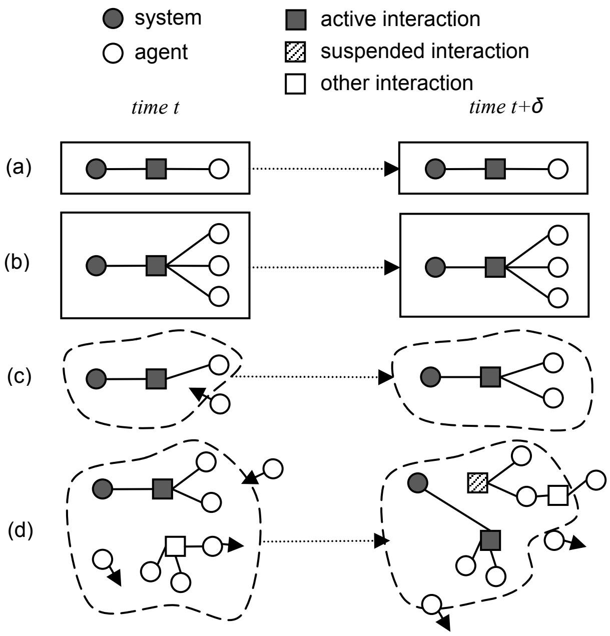


Figure 2. Conversational dynamics in: (a) single-user system; (b) a fixed multi-participant system; (c) an open multi-participant system, (d) an open multiparty system

lems, such as identifying the source and the target for each communicative signal in a multi-participant interaction, or handling engagement and disengagement in dynamic multi-participant settings. Moving from multi-participant to multiparty systems raises additional problems with respect to maintaining multiple interaction contexts, and triaging attention between multiple goals, parties and conversations. We shall discuss these new challenges in more detail in Section 3. Before that, we turn our attention to a second central feature of open-world dialog: the situated nature of the interaction.

2.2. Situated Aspect of Open-World Dialog

Dialog systems developed to date operate within narrow, predefined communication contexts. For example, in telephony-based spoken dialog systems, the audio-only channel limits the available context to the information that can be gained through dialog. In some cases, a stored user profile might provide additional information. Multimodal mobile systems might also leverage additional context from simple sensors like a GPS locator.

In contrast, systems designed to be effective in the open world will often need to make inferences about multiple aspects of the context of interactions by considering rich streams of evidence available in the surrounding environment. Such evidence can be observed by standing sensors or actively collected to resolve critical uncertainties. People are physical, dynamic entities in the world, and the system must reason about them as such, and about the conversational scene as a whole, in order to successfully and naturally manage the interactions. Concepts like presence, identity, location, proximity, trajectory, attention, and inter-agent relationships all play fundamental roles in shaping natural, fluid interactions, and need to become first-order objects in a theory of open-world dialog.

Like the multiparty aspect of open-world dialog, the situated nature of the interaction raises a number of new research challenges and brings novel dimensions to existing problems. One challenge is creating a basic set of physical and situational awareness skills. Interacting successfully in open environments requires that information from multiple sensors is fused to detect, identify, track and characterize the relevant agents in the scene, as well as the relationships between these agents. At a higher level, models for inferring and tracking the activities, goals, and long-term plans of these agents can provide additional context for reasoning within and beyond the confines of a given interaction, and optimizing assistance to multiple parties. Finally, new challenges arise in terms of integrating this streaming context in various interaction processes, like the engagement or disengagement process, turn taking, intention recognition, and multiparty dialog management.

3. Core Competencies for Open-World Dialog

We anchor our discussion of challenges for open-world dialog in Clark's model of language interaction [7]. With this model, natural language interaction is viewed as a joint activity in which participants in a conversation attend to each other and coordinate their actions on several different levels to establish and maintain mutual ground. Components of Clark's perspective are displayed in Figure 3. At the lowest level (*Channel*), the participants coordinate their actions to establish, maintain or break an open communication channel. At the second (*Signal*) level, participants coordinate the presentation and recognition of various communicative signals. At the third (*Intention*) level, participants coordinate to correctly interpret the meaning of these signals. Finally, at the fourth (*Conversation*) level, participants coordinate and plan their overall collaborative activities and interaction.

Successfully engaging in dialog therefore requires a minimal set of competencies at each of these levels. And indeed, most spoken dialog systems are organized architecturally in components that closely mirror Clark's proposed model: a voice activity detector and speech (and/or gesture) recognition engine identify the communicative signals, a language understanding component which extracts a corresponding semantic representation, and a dialog management component which plans the interaction.

We review in the rest of this section challenges raised by the multiparty and situated aspects of open-world dialog in each of these areas. We begin at the *Channel* level.

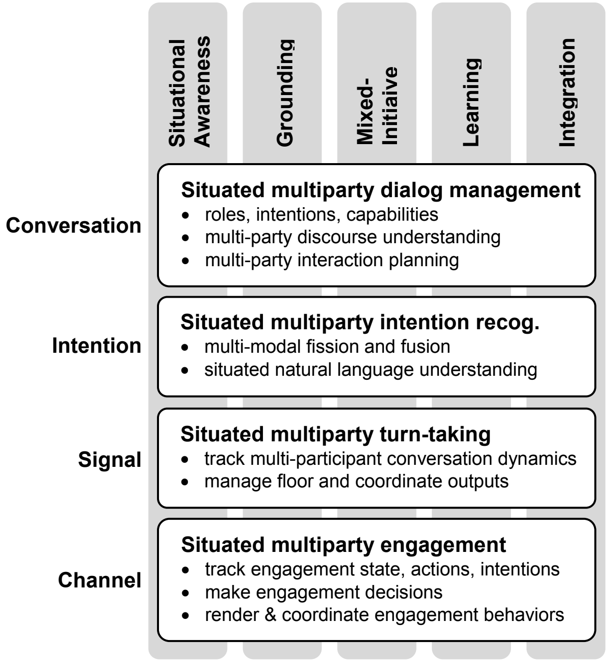

Situational Awareness
Grounding
Mixed-Initiaive
Learning
Integration
Conversation
Situated multiparty dialog management
• roles, intentions, capabilities
• multi-party discourse understanding
• multi-party interaction planning
Intention
Situated multiparty intention recog.
• multi-modal fission and fusion
• situated natural language understanding
Signal
Situated multiparty turn-taking
• track multi-participant conversation dynamics
• manage floor and coordinate outputs
Channel
Situated multiparty engagement
• track engagement state, actions, intentions
• make engagement decisions
• render & coordinate engagement behaviors

Figure 3. Core competencies for open-world dialog

3.1. Situated Multiparty Engagement

As a prerequisite for interaction, participants in a dialog must coordinate their actions to establish and maintain an open communication channel. In single-user systems this problem is often solved in a trivial manner. For instance, in telephony-based spoken dialog systems the channel is assumed to be established once a call has been received. Similarly, multimodal mobile applications oftentimes resolve the channel problem by using a push-to-talk solution.

Although these solutions are sufficient and perhaps natural in closed, single-user contexts, they become inappropriate for systems that must operate continuously in open, dynamic environments. We argue that such systems should ideally implement a *situated multiparty engagement model* that allows them to fluidly engage, disengage and re-engage in conversations with one or more participants.

Observational studies have revealed that humans negotiate conversational engagement via a rich, mixed-initiative, coordinated process in which non-verbal cues and signals, such as spatial trajectory and proximity, gaze and mutual attention, head and hand gestures, and verbal greetings all play essential roles [2, 3, 14]. Successfully modeling this coordinated process requires that the system (1) can sense and reason about the engagement actions, state and intentions of multiple agents in the scene, (2) can make high-level engagement control decisions (such as whom to engage with and when), and (3) can render engagement decisions in low-level coordinated behaviors and outputs.

Models for sensing the engagement state, actions, and intentions of various agents in the scene are, to a large extent, predicated on the system's capabilities to understand the physical environment in which it is immersed, *i.e.* to detect, identify and track multiple agents, including their location, trajectory, focus of attention, and other engagement cues. Higher-level inferences about the long-term goals, plans and activities of each agent can also provide informative priors for detecting engagement actions.

Beyond the engagement sensing problem, at a higher level, the system must reason about the boundaries of each conversation and make real-time decisions about whom to engage (or disengage) with, and when. In a dynamic multiparty setting these decisions have to take into account additional streams of evidence, and optimize tradeoffs between the goals and needs of the multiple parties involved (*e.g.,* interrupting a conversation to attend to a more urgent one). In making and executing these decisions, the system must consider social and communicative expectations and etiquette. Finally, such high-level engagement decisions must be signalled in a meaningful, understandable manner to the relevant participants. For instance, in an embodied anthropomorphic agent, engagement actions have to be rendered into a set of corresponding behaviors (*e.g.*, establishing or breaking eye contact, changing body posture, generating subtle facial expressions, or issuing greetings) that must often be coordinated at the millisecond scale.

3.2. Situated Multiparty Turn Taking

Going one level up in Clark's model, at the *Signal* level, the system must coordinate with other participants in the conversation on the presentation and recognition of communicative signals (both verbal and non-verbal, *e.g.,* gestures and emotional displays.) The coordinated process by which participants in a conversation take turns to signal to each other is known as turn-taking and has been previously investigated in the conversational analysis and psycholinguistics communities, *e.g.* [12, 18]. While computational models for turn-taking [19, 23, 24] have also been proposed and evaluated to date, most current systems make simplistic one-speaker-at-a-time assumptions and have relied on voice activity detectors to identify when the user is speaking. Phenomena like interruptions or barge-ins are often handled using ad-hoc, heuristic solutions, which can lead to turn-overtaking issues and ultimately to complete interaction breakdowns even in single-user systems [6].

Open-world dialog requires the development of a computational, *situated multiparty turn-taking model.* On the sensing side, such a model should be able to track the multi-participant conversational dynamics in real time by fusing lower-level evidence streams (*e.g.,* audio and visual). The model should be able to identify the various communicative signals as they are being produced, and, in a multi-participant setting, identify the sender, the addressees (and potentially the over-hearers) for each signal. In addition, the model should be able to track who has the conversational floor, i.e., the right to speak, at any given point in time. On the control side, a multiparty situated turn-taking model should make real-time decisions (that are in line with basic conversational norms) about when the system can or should start or stop speaking, take or release the conversational floor, etc. Finally, the model must coordinate the system's outputs and render them in an appropriate manner. For instance, in an embodied conversational system, speech, gaze, and gesture must be tightly coordinated to signal that the system is addressing a question to two conversational participants, or to indicate that the system is trying to currently acquire the floor.

3.3. Situated Multiparty Intention Recognition

At the *Intention* level, a dialog system must correctly interpret the meaning of the identified communicative signals. In traditional dialog systems this is the realm of the language understanding component. Given the static, relatively limited communication context, the language understanding challenges tacked in traditional dialog systems have been typically limited to generating an appropriate semantic representation for the hypotheses produced by a speech recognizer, and integrating this information with the larger dialog context. In certain domains, issues like ellipsis and anaphora resolution also have played an important role. Systems that use multiple input modalities (*e.g.,* speech and gesture) face the problem of multi-modal fusion at this level: signals received from the lower levels must be fused based on content and synchronicity into a unified semantic representation of the communicative act.

The physically situated nature of open-world dialog adds new dimensions to each of these problems. In situated interactions, the surrounding environment provides rich streaming context that can oftentimes be leveraged for intention recognition. For instance, in the receptionist domain, an interactive system might be able to infer intentions based on identity (John always needs a shuttle at 3pm on Wednesday), spatiotemporal trajectories (people entering the lobby from inside the building are more likely to want a shuttle reservation than people entering the lobby from outside the building), clothing and props (a formally-dressed person is more likely a visitor who wants to register than an internal employee), and so on. Novel models and formalisms for reasoning about the streaming context and fusing it with the observed communicative signals to decode intentions and update beliefs are therefore required.

An additional challenge for open-world dialog is that of situated language understanding. Physically situated systems might often encounter deictic expressions like "Come here!" "Bring me the red mug," and "He's with me", etc. Resolving these referring expressions requires a set of language understanding skills anchored in spatial reasoning and a deep understanding of the relevant entities in the surrounding environment and of the relationships between these entities. The same holds true for pointing gestures and other non-verbal communicative signals.

3.4. Situated Multiparty Dialog Management

At the fourth level, referred as the *Conversation* level, participants coordinate the high-level planning of the interaction. This is the realm of dialog management, a problem that has already received significant attention in the spoken dialog systems community, *e.g.* [4, 6, 8, 16, 17, 20]. However, with the exception of a few incipient efforts [15, 25], current models make an implicit single-user assumption, and do not deal with the situated nature of the interactions.

One of the main challenges for open-world spoken dialog systems will be the development of models for *mixed-initiative, situated multiparty dialog management.* To illustrate the challenges in this realm, consider the situation in which a visitor, accompanied by her host, engages in dialog with a receptionist to obtain a visitor's badge. In order to successfully plan multi-participant interactions, the dialog manager must model and reason about the goals and needs of different conversational partners (*e.g.* get a badge versus accompany the visitor), their particular roles in the conversation (*e.g.* visitor versus host), their different knowledge and capabilities (*e.g.* only the visitor knows the license plate of her car). Individual contributions, both those addressed to the system, and those that the participants address to each other, need to be integrated with a larger multi-participant discourse and situational context.

Mixed-initiative interaction [10] with multiple participants requires that the system understands how to decompose the task at hand, and plan its own actions accordingly (*e.g.* directing certain questions only to certain participants, etc.) All the while, the dialog planning component must be able to adapt to the dynamic and asynchronous nature of the open-world. For instance, if the visitor's host disengages momentarily to greet a colleague in the lobby, the system must be able to adjust its conversational plans on-the-fly to the current situation (*e.g.* even if it was in the middle of asking the host a question at that point)

Handling multiparty situations (*e.g.* a third participant appears and engages on a separate topic with the host) requires that the system maintain and track multiple conversational contexts, understand potential relationships between these contexts, and is able to switch between them. Furthermore, providing long-term assistance requires that the system is able to reason about the goals, activities and long-term plans of individual agents beyond the temporal confines of a given conversation. To illustrate, consider another example from the receptionist domain: after making a reservation, a user goes outside to wait for the shuttle. A few minutes later the same user re-enters the building and approaches the reception desk. The receptionist infers that the shuttle probably did not arrive and the user wants to recheck the estimated time of arrival or to make another reservation; she glances towards the user and says "Two more minutes." Inferences about the long-term plans of various agents in the scene can provide valuable context for the streamlining the interactions.

3.5. Other Challenges

So far, we have made use of Clark's four-level model of grounding to identify and discuss a set of four core competencies for open-world spoken dialog systems: multiparty situated engagement models, multiparty situated turn-taking models, situated intention recognition, and mixed-initiative multiparty dialog management. However, developing an end-to-end system requires more than a set of such individual models. A number of additional challenges cut across each of these communicative processes. In the remainder of this section, we briefly review five challenges: situational awareness, robustness and grounding, mixed-initiative interaction, learning, and integration.

Given the situated aspect of open-world interaction, a major overarching challenge for open-world spoken dialog systems is that of *situational awareness*. As we have already seen, the ability to fuse multiple sensor streams and construct a coherent picture of the physical surrounding environment and of the agents involved in the conversational scene plays a fundamental role in each of the conversational processes we have previously discussed. Open-world systems should be able to detect, identify, track and characterize relevant agents, events, objects and relationships in the scene. Models for reasoning about the high-level goals, intentions, and long-term plans of the various agents can provide additional information for establishing rapport and providing long-term assistance. In contrast to traditional work in activity recognition (*e.g.,* in the vision or surveillance community), interactive systems also present opportunities for eliciting information on the fly and learning or adapting such models through interaction.

A second major challenge that spans the communicative processes discussed above is that of dealing with the uncertainties resulting from sensor noise and model incompleteness. Uncertainties abound even in human-human communication, but we are generally able to monitor the conversation and re-establish and maintain mutual ground. Open-world dialog systems can benefit from the development of similar *grounding models* that explicitly represent and make inferences about uncertainties at different levels and, when necessary, take appropriate actions to reduce the uncertainties and re-establish mutual ground.

A third important overall challenge is that of *mixed-initiative interaction***.** So far, we have discussed the notion of mixed-initiative in the context of the dialog management problem. It is important to notice though that, like situational awareness and grounding, the notion of mixed-initiative pervades each of the communicative processes we have discussed. At each level, the system's actions need to be tightly coordinated with the actions performed by the other agents involved in the conversation. Examples include the exchange of cues for initiating or breaking engagement, or "negotiating" the conversational floor. Mechanisms for reasoning about and managing initiative will therefore play a central role in each of these layers.

A fourth important challenge that cuts across the four competencies discussed above is that of *learning*. Given the complexities involved, many of the models we have discussed cannot be directly authored but must be learned from data. Ideally, we would like to build systems that learn throughout their lifetimes, directly from interaction, from their experience, without explicit supervision from their developers. Furthermore, such systems should be able to share the knowledge they acquire with each other.

Finally, another challenge not be underestimated is that of *system integration*, of weaving together all these different components into an architecture that is transparent, modular, and operates asynchronously and in real-time to create a seamless natural language interaction.

4. A Prototype System

We now describe a concrete implementation of a prototype system, named the Receptionist. The Receptionist is a situated conversational agent that can fluidly engage with one or more people and perform tasks typically handled by front-desk receptionists (*e.g.,* making shuttle reservations, registering visitors, providing directions on campus, etc.) at our organization. In previous work in this domain [11], we have investigated the use of a hierarchy of Bayesian models and decision-theoretic strategies for inferring intentions and controlling question asking and backtracking in dialog. Here, we focus on exploring the broader challenges of open-world dialog.

The front-desk assistance domain has several properties that make it a valuable test-bed for this endeavor. The interactions happen in an open, public space (building lobbies) and frequently involve groups of people. The complexity of the tasks involved ranges from the very simple, like making shuttle reservations, to more difficult ones requiring complex collaborative problem solving skills. Finally, a deployed system could provide a useful service and its wide adoption would create a constant stream of ecologically-valid real-world interaction data.

In the rest of this section, we describe the Receptionist system, and discuss an initial set of models that address the core competencies for open-world dialog we have previously outlined. In particular, we focus our attention on the situational awareness, engagement, and multi-participant turn-taking capabilities of this system. Despite the preliminary and sometimes primitive nature of these models (they represent only a first iteration in this long-term research effort), as we shall see in Section 5, when weaved together, they showcase the potential for seamless natural language interaction in open, dynamic environments.

We begin with a high-level overview of the hardware and software architecture. The current prototype takes the form of an interactive multi-modal kiosk, illustrated in Figure 4. On the input side, the system uses four sensors: a wide-angle camera with 140° field of view and a resolution

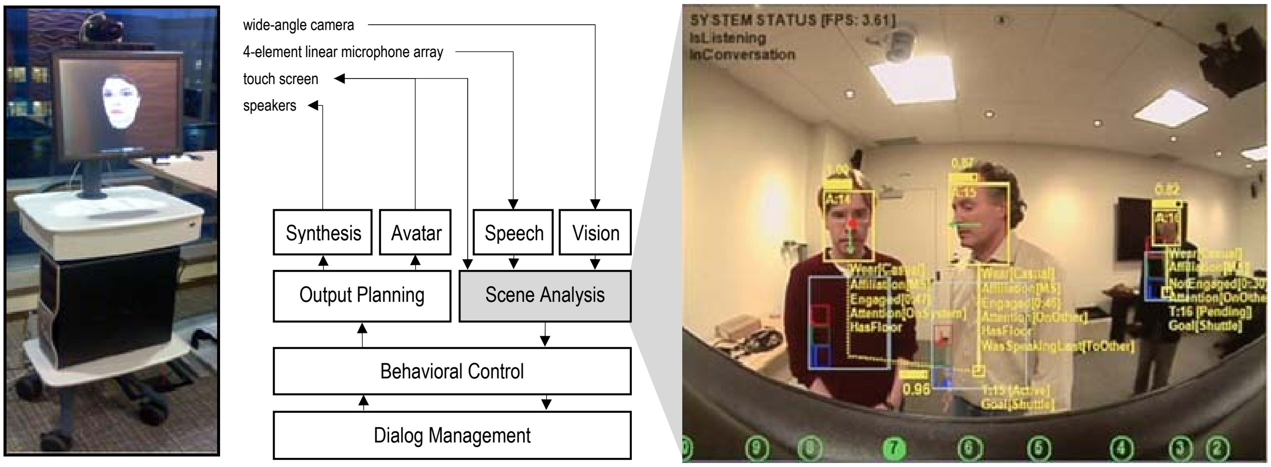


Figure 4. Receptionist system: (a) prototype, (b) architectural overview, and (c) runtime conversational scene analysis

of 640x480 pixels; a 4-element linear microphone array that can provide sound-source localization information in 10° increments; a 19" touch-screen; and a RFID badge reader. As output, the system displays a realistic talking avatar head, which is at times complemented by a graphical user interface (*e.g.* when speech recognition fails the GUI is displayed and users can interact via the touch-screen – see Figure 5.c). The system currently runs on a 3.0GHz dual-processor Intel Xeon machine (total 8 cores).

Data gathered by the sensors is forwarded to a scene analysis module that fuses the incoming streams and constructs (in real-time) a coherent picture of what is happening in the surrounding environment. This includes detecting and tracking the location of multiple agents in the scene, reasoning about their attention, activities, goals and relationships (*e.g.* which people are in a group together), and tracking the current conversational context at different levels (*e.g.* who is currently engaged in a conversation, who is waiting to engage, who has the conversational floor, who is currently speaking to whom, etc.) The individual models that implement these functions are described in more detail in the sequel.

The conversational scene analysis results are then forwarded to the control level, which is structured in a two-layer reactive-deliberative architecture. The lower-level, reactive layer implements and coordinates various low-level behaviors (*e.g.* for engagement and conversational floor management, for coordinating spoken and gestural outputs, etc). The higher-level, deliberative layer makes conversation control decisions, planning the system dialog moves and high-level engagement actions.

4.1. Situational Awareness

The system currently implements the following situational awareness capabilities.

Face detection and tracking. A multiple face detector and tracker are used to detect and track the location $x_a(t)$ of each agent a in the scene. The face detector runs at every frame and is used to initialize a mean-shift tracker. The frame-to-frame face correspondence problem is resolved by a proximity-based algorithm. These vision algorithms run on a scaled-up image (1280x960 pixels), which allows us to detect frontal faces up to a distance of about 20 feet. Apart from the face locations $x_a(t)$ and sizes $w_a(t)$, the tracker also outputs a face confidence score $fc_a(t)$, which is used to prune out false detections but also to infer focus of attention (described later.)

Pose tracking. While an agent is engaged in a conversation with the system, a face-pose tracking algorithm runs on a cropped region of interest encompassing the agent's face. In group conversations, multiple instances of this algorithm run in parallel on different regions of interest. The pose tracker provides 3D head orientation information for each engaged agent $\overline{\omega_a}(t)$, which is in turn used to infer the focus of attention (see below.)

Focus of attention. At every frame, a direct conditional model is used to infer whether the attention of each agent in the scene is oriented towards the system or not: $P(foa_a(t)|fc_a(t),\overline{\omega_a}(t))$. This inference is currently based on a logistic regression model that was trained using a hand-labelled dataset. The features used are the confidence score from the face tracker $fc_a(t)$ (this is close to 1 when the face is frontal), and the 3D head orientation generated by the pose tracker $\overline{\omega_a}(t)$, when available (recall that the pose tracker runs only for engaged agents.)

Agent characterization. In addition to face detection and tracking, the system also performs a basic visual analysis of the clothing for each detected agent. The probability that the agent is formally or casually dressed $P(formal_a(t))$ is estimated based on the color variance in a rectangular patch below the face (*e.g.* if a person is wearing a suit, this typically leads to high variance in this image patch). This information is further used to infer the agent's likely affiliation, based on a simple conditional model $P(affiliation_a(t)|formal_a(t))$. Casually dressed agents are more likely to be Microsoft employees; formally dressed ones are more likely to be visitors.

Group inferences. Finally, the Receptionist system also performs a pairwise analysis of the agents in the scene to infer group relationships. The probability of two agents being in a group together $P(group(a_1, a_2))$ is computed by a logistic regression model that was trained on a hand-labelled dataset. The model uses as features the size, location and proximity of the faces, but also observations collected through interaction. For instance, the system might ask a clarification question like "Are the two of you together?" A positive or negative response to this question is also used as evidence by the group inference model.

4.2. A Multiparty Situated Engagement Model

We now turn our attention to the problem of *engagement* [21], the process by which participants in a conversation establish, maintain and terminate their interactions (corresponding to the first level of coordinated action in Clark's language interaction model).

The engagement model currently used in the Receptionist system is centered on a reified notion of *interaction*, defined here as a basic unit of sustained, interactive problem-solving. Each interaction involves two or more participants, and this number may vary in time: new participants may join an existing interaction, and current participants may leave an interaction. The system is actively engaged in at most one interaction at a time, but it can simultaneously keep track of additional, suspended interactions. Engagement is then viewed as the joint activity of the system and its users by which interactions are initiated, terminated, suspended, resumed, joined or abandoned.

To manage this coordinated process, the system: (1) constantly monitors the engagement state, actions and intentions of surrounding agents, (2) makes high-level decisions about whom to engage (or disengage) with and when, and (3) renders these decisions via behaviors such as establishing or breaking eye contact, issuing and responding to verbal greetings, etc. In the following subsections, we discuss each of these components in more detail.

4.2.1. Engagement State, Actions, and Intentions

The basis for making engagement decisions is provided by a model that tracks the engagement state $ES_a(t)$, actions $EA_a(t)$ and intentions $EI_a(t)$ for each agent in the scene.

The engagement state of an agent $ES_a(t)$ is modeled as a deterministic variable with two possible values: *engaged* and *not-engaged*, and is updated based on the joint actions of the agent and the system. The state transitions to *engaged* when both the system and an agent take an engaging action. On the other hand, disengagement can be a unilateral act: if either the system or an engaged agent take a disengaging action, the state transitions to *not-engaged*.

A second engagement variable, $EA_a(t)$, models the actions that an agent takes to initiate, maintain and terminate engagement (i.e. to transition between engagement states). There are four possible engagement actions: *engage*, *no-action*, *maintain*, *disengage*. An agent can take the first two actions only from the *not-engaged* state and the last two only from the *engaged* state. Currently, a direct conditional model $P(EA_a(t)|ES_a(t), \Psi(\mathrm{t}))$ is used to estimate an agent's engagement action based on the current engagement state and additional evidence $\Psi(\mathrm{t})$ gathered from various sensors and processes in the system. Examples include the detection of greetings or calling behaviors (*e.g.* "Hi!" or "Laura!"), the establishment or the breaking of a conversation frame (*e.g.* the agent approaches and positions himself in front of the system; or the agent departs), continued attention (or lack thereof) to the system, etc.

Apart from the engagement state and actions, the system also keeps track of a third variable, the engagement intention $EI_a(t)$ of each agent in the scene; this can be *engaged* or *not-engaged*. Intentions are tracked separately from actions since an agent might intend to engage the system, but not take a direct, explicit engagement action. A typical case is that in which the system is already engaged in an interaction and the participant is simply waiting in line. More generally, the engagement intention corresponds to whether or not the user would respond positively should the system initiate engagement. Currently, the engagement intention is inferred using a handcrafted direct conditional model $P(EI_a(t)|ES_a(t), EA_a(t), \Psi(t))$ that leverages information about the current engagement state and action, as well as additional evidence gleaned from the scene including the spatiotemporal trajectory of the participant, the level of sustained mutual attention, etc.

While the current models for sensing engagement actions and intentions are handcrafted, we are also investigating data-driven approaches for learning these models.

4.2.2. Engagement Decisions

Based on the inferred state, actions and intentions of the agents in the scene, as well as other additional evidence, the system makes high-level decisions about when and with whom to engage in interaction. The system's engagement action-space at contains the same four actions previously discussed. The actual surface realization of these actions in terms of low-level behaviors, such as greetings, making or breaking eye contact, etc. is discussed in more detail in the following subsection.

As the Receptionist system operates in an open, multiparty environment, the engagement decisions can become quite complex. For instance, new participants might arrive and wait to engage while the system is already engaged in an interaction; in some cases, they might even actively try to barge-in and interrupt the current conversation. In such cases, the system must reason about the multiple tasks at hand, and balance the goals and needs of multiple participants in the scene and resolve various trade-offs, for instance between continuing the current interaction and temporarily interrupting it to address a new (perhaps shorter and more urgent task).

Currently, a simple heuristic model is used for making these decisions. If the system is not currently engaged in an interaction, it conservatively waits for a user to initiate engagement (*e.g.* $EA_a(t)$=*engage*), before making the decision to engage. In addition, if the system is currently engaged in a conversation interaction, but other agents are present and waiting to engage (*e.g.* $EI_a(t)$=*engaged*, $EA_a(t)$=*no-action*), the system may suspend the current interaction to momentarily engage a waiting agent to either let them know that they will be attended to momentarily, or to inquire about their goals (this is illustrated in more detail in Section 5.) This decision is made by taking into account the appropriateness of suspending the current conversation at that point, and the waiting time of the agent in the background. We are currently exploring more principled models for optimizing the scheduling of assistance to multiple parties under uncertainties about the estimated goals and needs, the duration of the interactions, time and frustration costs, social etiquette, etc.

4.2.3. Engagement Behaviors

Each high-level engagement decision (*e.g. Engage* / *Disengage*) is rendered into a set of coordinated lower-level behaviors, such as making and breaking eye contact, issuing greetings, etc.

The sequencing of these lower-level behaviors is highly dependent on the current situation in the scene, including the estimated engagement state, actions and intentions for each agent, the evolving state of the environment and system (*e.g.* is the system in a conversation or not, are there other agents in the scene, what is their focus of attention, etc.) For instance, consider the case when the system is not yet engaged in any conversations and a high-level decision is made to engage a certain agent. If mutual attention has already been established, the *engage* behavior triggers a greeting. In contrast, if the agent's focus of attention is not on the system, the *engage* behavior attempts to draw the agent's attention by gazing towards him or her and saying "Excuse me!" in a raised voice. After the initial salutation the system monitors the spatiotemporal trajectory of the agent, and, if the agent approaches the system, establishes or maintains mutual attention, the *engage* behavior completes successfully; the agent's engagement state is updated to *engaged*. Alternatively if a period of time elapses and the agent does not establish mutual attention (or leaves the scene), the *engage* behavior completes with failure (which is signalled to the higher engagement control layer). The system implements several other engagement and disengagement behaviors dealing with agents joining or leaving an existing conversation. While a full description of these behaviors is beyond the scope of this paper, instances of various engagement behaviors are illustrated in the example discussed in Section 5.

4.3. Multi-Participant Turn Taking

While engaged in a conversation, the system coordinates with other conversational participants on the presentation and recognition of various communicative signals. Our current prototype attends to verbal signals (*i.e.,* spoken utterances) and to signals received from the graphical user interface, which can be accessed via the touch-screen. On the output side, the system coordinates spoken outputs with gaze and various gestures such as smiles, and furrowed or questioning eye-brows.

A voice activity detector is used to identify and segment out spoken utterances from background noise. The speaker S_u for each utterance u is identified by a model that integrates throughout the duration of the utterance the sound source localization information provided by the microphone array with information from the vision subsystem, specifically the location of the agents in the scene. For each identified utterance, the system infers whether the utterance was addressed to the system or not. This is accomplished by means of a model that integrates over the user's inferred focus of attention throughout the duration of the spoken utterance $P(T_u = system|foa_{S_u}(t))$. If the user's focus of attention stays on the system, the utterance is assumed to be addressed to the system; otherwise, the utterance is assumed to be directed towards the other participants engaged in the conversation. Touch events detected by the graphical user interface are assumed to be generated by the closest agent, and addressed to the system.

In order to fluidly coordinate its own outputs (*e.g.* spoken utterances, gestures, GUI display) with the other agents engaged in the conversation, the system implements a simple multiparty situated turn-taking model. The model tracks whether or not each engaged agent currently holds the conversational floor $FS_a(t)$ (*i.e.* has the right to speak), and what the floor management actions each engaged agent takes at any point in time $FA_a(t)$: *No-Action*, *Take-Floor*, *Release-to-System*, *Release-to-Other*, *Hold-Floor*. These actions are inferred based on a set of handcrafted rules that leverage information about the current state of the floor $\{FS_a(t)\}_a$, the current utterance u, its speaker S_u and its addressees T_u. For instance, a *Take-Floor* action is detected when a participant does not currently hold the floor but starts speaking or interacts with the GUI; a *Release-to-System* action is detected when a participant finishes speaking, and the utterance was addressed to the system; and so on. The floor state for each agent $FS_a(t)$ is updated based on the joint floor-management actions of the system and engaged agents. For instance if a user currently holds the floor and performs a *Release-to-System* action, immediately afterwards the floor is assigned to the system.

Based on who is currently speaking to whom and on who holds the floor, the system coordinates its output with the other conversational participants. For instance, the system behavior that generates spoken utterances verifies first that the system currently holds the floor. If this is not true, a floor management action is invoked for acquiring the floor. The lower level behaviors render this action by coordinating the avatar's gaze, gesture and additional spoken signals (*e.g.* "Excuse me!", if the system is trying to take the floor but a participant is holding it and speaking to another participant).

The current multi-participant turn-taking model is an initial iteration. It employs heuristic rules and limited evidential reasoning, treats each participant independently, and does not explicitly take into account the rich temporality of interactions. We are exploring the construction and use of more sophisticated data-driven models for jointly tracking through time the speech source S_u, target T_u, focus of attention $foa_a(t)$ and floor state $FS_a(t)$ and actions $FA_a(t)$ in multi-participant conversation, by fusing through time audio-visual information with additional information about the system actions (*e.g.* its pose and gaze trajectory, etc.) and the history of the conversation: $P(S_u, T_u, foa_{\{a\}}(t), FS_{\{a\}}(t), FA_{\{a\}}(t)|\Psi(t))$

4.4. Situated Intention Recognition

To infer user goals and intentions, the Receptionist system makes use of several hybrid belief updating models that integrate streaming evidence provided by the situational context, with evidence collected throughout the dialog. For instance, the system relies on a conditional goal inference model $P(G_a|affiliation_a, group(a, a_i), SG_a)$ that currently takes that takes into account the estimated actor affiliation and whether or not the actor is part of a larger group (e.g. Microsoft employees are more likely to want shuttles than to register as visitors, people in a group are more likely to register as visitors, etc.) If the probability of the most likely goal does not exceed a grounding threshold, the system collects additional evidence - SG_a - through interaction, by directly asking or confirming the speculated goal. Similarly, in case an agent's goal is to make a shuttle reservation, the number of people for the reservation is inferred by a model that integrates information from the scene (*e.g.* how many people are present) with data gathered through dialog. The runtime behavior of these models is illustrated in more detail in the following section.

5. A Sample Interaction

We now illustrate how the models outlined in the previous section come together to create a seamless multiparty situated interaction, by describing a sample interaction with the receptionist system. Figure 5 shows several successive snapshots from a recorded interaction, with the runtime annotations created by the various models, as well as a capture of the system's display and a transcript of the conversation. A full video capture is available online [22].

Initially two participants are approaching the system (A14 and A15 in Figure 5). The system detects and tracks their location. As the users get closer and orient their attention towards the system, the engagement model indicates that they are performing an engaging action. In response, the avatar triggers an engaging behavior, greets them and introduces itself (line 3 in Figure 5).

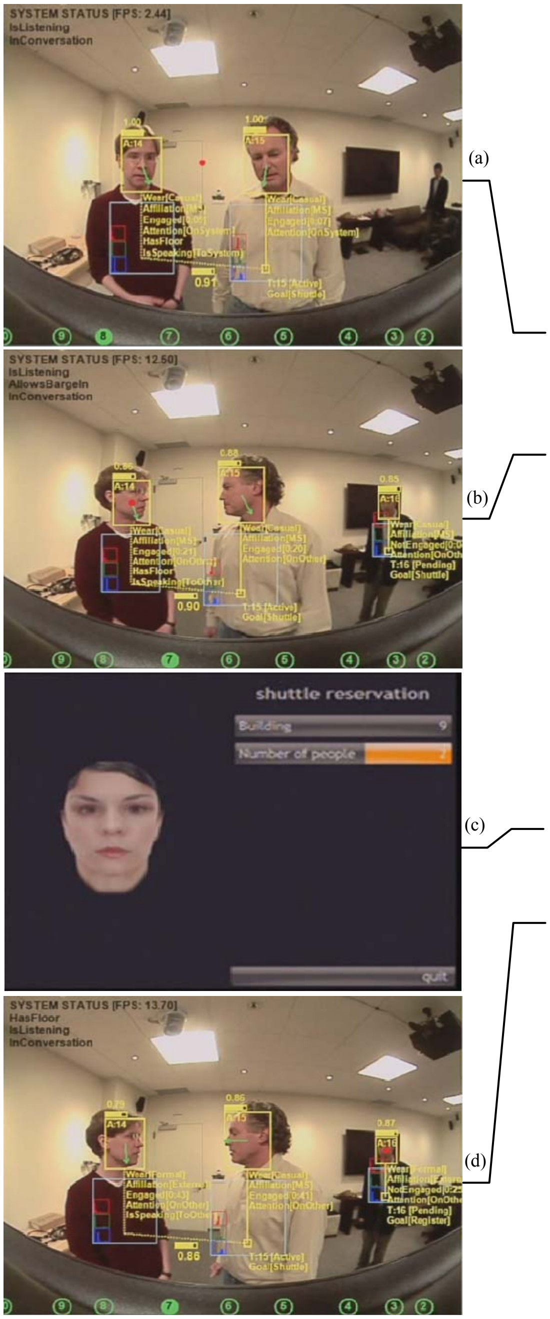
After the initial greeting, the system attempts to ground the goals of the two participants. The group inference model indicates that, with high likelihood (0.91 in Figure 5.a) the two participants are in a group together. The clothing and affiliation models indicate that the two participants and dressed casually, and therefore most likely Microsoft employees. Based on this information, the system infers that the participants most likely want a shuttle. Since the likelihood of the shuttle goal does not exceed the grounding threshold, the system confirms this information through dialog, by glancing at the two participants and asking: "Do you need a shuttle?" A14 confirms.

Next, the system asks "Which building are you going to?" At this point (see also Figure 5.b) the first participant (A14) turns towards the second one (A15) and initiates a side conversation (lines 8-12). By fusing information from the microphone array, the face detector and pose tracker, the multiparty turn-taking model infers that the two participants are talking and releasing the floor to each other. Throughout this side conversation (lines 8-12) the avatar's gaze follows the speaking participant. In addition, the recognition system is still running and the system overhears the building number from this side conversation. When the two participants turn their attention again towards the system, the turn-taking model identifies a *Release-To-System* floor action. At this point, the system continues the conversation by confirming the overheard information: "So you're going to 9, right?" A14 confirms again.

Next, the system grounds how many seats are needed for this reservation. Here, a belief updating model fuses information gathered from the scene analysis with information collected through interaction. Based on the scene, the system infers that most likely this shuttle reservation is for two people (A14 and A15). The likelihood however does not exceed a grounding threshold (since at this point a third agent has already appeared in the background – A16). The system therefore confirms the number of seats through dialog, by asking "And this is for both of you, right?" Once the number of people is grounded, the system notifies A14 and A15 that it is currently making a reservation for them.

As we have already noted, while A14 and A15 where engaged in the side conversation (lines 8-12), a new participant (A16) entered the scene – see Figure 5.b. When the new participant appears, the system glances for a fraction of a second at him (this is a hard-coded reactive behavior). The group models indicate that A16 is most likely not in a group with A14 and A15. The clothing and affiliation models for A16 indicate that this participant is dressed formally and therefore most likely to be an external visitor. As a consequence, the activity and goal models indicate that A16 is waiting for the receptionist with the intention to register.

After the avatar notifies A14 and A15 that it is making their shuttle reservation, these two participants turn again to each other and begin another side conversation. The system decides to temporarily suspend its conversation with



1 A14, A15: *[approaching, talking to each other]*
2 A14, A15: *[getting close and turning their attention to the system]*
3 S: Hi, my name is Laura and today I'm here to help the receptionist with shuttle reservation
4 A14: I'm Dan and this is Eric
5 S: *[glancing at both participants]*
Nice to meet you. Do you need a shuttle?
6 A14: Yes
7 S: *[turns to A14]*
Which building are you going to?
8 A14: *[turns to A15]*
I forget where are we going?
9 S: *[quickly glances at A16, who just entered the scene; then follows the side conversation that A14 and A15 are having, by turning its attention towards the speaking participant]*
10 A15: *[turns to A14]*
I think it's building 9?
11 A14: *[still looking at A15]*
Are you sure?
12 A15: *[still looking at A14]*
Yes
13 A14,A15: *[both turn back towards the system]*
14 S: *[glancing at both A14 and A16]*
So you're going to building 9, right?
15 A14: Yes
16 S: And this if for both of you, right?
17 A14: Yes
18 S: I'm making a shuttle reservation for building 9 for 2 people. In case you want to correct anything, say or press start over.
19 S: *[turning to A16]*
Are you visiting someone?
20 A16: Yes
21 S: I'll help you register in just a moment.
22 S: *[turning back to A14 and A15]*
Excuse me.
23 A14,A15: *[turning their attention back to the system]*
24 S: You will be on shuttle 53. It should be here any second.
25 A14: All right, thank you. Bye
26 S: *[glancing at A14 and A15]*
Bye-bye
27 A16: *[approaching]*
Hi

Figure 5. Sample interaction with Receptionist.

A14 and A15 and engages with A16 by asking "Are you visiting someone?" A16 confirms, and the system notifies A16 that it will help with the registration momentarily. The decision to suspend the conversation with A14 and A15 is taken by the high-level engagement control model, which is aware of the fact that the current conversation is interruptable at this point (the system is waiting for the shuttle backend to respond and A14 and A15 are talking to each other), and that, with high likelihood, there is an additional participant in the scene (A16) waiting for assistance.

After the system notifies A16 that it would attend to his needs momentarily (line 22), the shuttle backend responds with the initial reservation. The system turns its attention again at A14 and A15, and attempts to resume that conversation, by invoking a corresponding engagement behavior. Since the two participants are still talking to each other and not paying attention to the system, the *Resume-Conversation* behavior triggers an "Excuse me!" prompt (line 22). As soon at A14 and A15's attention turns back to the system, the avatar provides the information about the shuttle number and estimated time of arrival (line 24). The two participants then disengage and the system turns its attention back to and engages with A16.

Conclusion and Future Work

We have outlined a research agenda aimed at developing computational systems that can interact naturally and provide assistance with problem-solving needs over extended periods of time in open, relatively unconstrained environments. We first introduced the pursuit and challenges of developing systems competent in *open-world dialog*—with the ability to support conversation in an open-world context, where multiple people with different needs, goals, and long-term plans may enter, interact, and leave an environment, and where the physical surrounding environment typically provides streaming evidence that is important for organizing and conducting the interactions.

The dynamic, multiparty and situated nature of open-world dialog brings new dimensions to traditional spoken dialog problems, like turn-taking, language understanding and dialog management. We found that existing models are limited in that they generally make an implicit single-user assumption and are not equipped to leverage the rich streaming context available in situated systems. Open-world settings pose new problems like managing the conversation engagement process in a multiparty setting, scheduling assistance to multiple parties, and maintaining a shared frame that includes inferences about the long-term plans of various agents--inferences that extend beyond the confines of an acute interaction.

To provide focus as well as an experimental testbed for the research agenda outlined in this paper, we have developed a prototype system that displays several competencies for handling open-world interaction. The prototype weaves together a set of early models addressing some of the open-world dialog challenges we have identified, and showcases the potential for creating systems that can interact with people on problem-solving needs with the ease and etiquette expected from a human.

We take the research agenda and the prototype described in this paper as a starting point. We plan to investigate the challenges we have outlined, and to develop and empirically evaluate computational models that implement core competencies for open-world dialog. We hope others will join us on the path towards a new generation of interactive systems that will be able embed interaction and computation deeply into the natural flow of daily tasks, activities and collaborations.

ACKNOWLEDGMENTS

We would like to thank George Chrysanthakopoulos, Zicheng Liu, Tim Paek, Qiang Wang, Cha Zhang for their contributions, useful discussions, and feedback.

REFERENCES

[1] A. Acero, N. Bernstein, R. Chambers, Y-C Ju, X. Li, J. Odell, P. Nguyen, O. Scholtz, G. Zweig. Live Search for Mobile: Web Services by Voice on the Cellphone. in Procs ICASSP'08. Las Vegas (2008)

[2] M. Argyle. Bodily Communication, International University Press, Inc, New York (1975).

[3] M. Argyle, and M. Cook. Gaze and Mutual Gaze, Cambridge University Press, New York, (1976)

[4] D. Bohus and A. Rudnicky. The RavenClaw Dialog Management Framework: Architecture and Systems, Computer Speech and Language, DOI:10.1016/j.csl.2008.10.001

[5] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, and H. Yan. Embodiment in Conversational Interfaces: Rea, in Procs of CHI'99, Pittsburgh, PA, (1999).

[6] R. Cole. Tooles for Research and Education in Speech Science, in Procs of International Conference of Phonetic Sciences, San Francisco, CA (1999)

[7] H.H. Clark, and E.F. Schaefer. Contributing to Discourse. Cognitive Science. 13. (1989)

[8] G. Ferguson, and J. Allen. TRIPS: An Intelligent Integrated Problem-Solving Assistant, in Procs of AAAI'98, Madison, WI (1998)

[9] J. Gustafson, N. Lindberg, and M. Lundeberg. The august spoken dialogue system, in Procs. Eurospeech'99, Budapest, Hungary (1999).

[10] E. Horvitz. Reflections on Challenges and Promises of Mixed-Initiative Interaction, in *AI Magazine* vol. 28, Number 2 (2007)

[11] E. Horvitz and T. Paek. A Computational Architecture for Conversation, in Procs of 7th International Conference on User Modeling, Banff, Canada (1999)

[12] J. Jaffe and S. Feldstein. Rhythms of Dialogue, Academic Press (1970)

[13] M. Johnston, S. Bangalore. MATCHKiosk: a multimodal interactive city guide, in Procs of ACL'04, Barcelona, Spain (2004).

[14] A. Kendon. Conducting Interaction: Patterns of Behavior in Focused Encounters, Studies in International Sociolinguistics, Cambridge University Press (1990)

[15] F. Kronlid. Steps towards Multi-Party Dialogue Management, Ph.D. Thesis, University of Gothenburg (2008)

[16] S. Larsson. Issue-based dialog management, Goteborg University, Ph.D. Thesis (2002)

[17] M. McTear. Spoken dialogue technology: enabling the conversational user interface, in ACM Computing Surveys 34(1):90-169.

[18] H. Sacks, A. Schegloff, G. Jefferson. A simplest systematic for the organization of turn-taking for conversation. *Language*, 50(4):696-735 (1974).

[19] A. Raux and M. Eskenazi. Optimizing Endpointing Thresholds using Dialogue Features in a Spoken Dialogue System, in Procs SIGdial'08, Columbus, OH (2008)

[20] C. Rich, C. Sidner, and N. Lesh. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction, in AI Magazine. 22:15-25 (2001)

[21] C. Sidner and C. Lee. Engagement rules for human-robot collaborative interactions, in IEEE International Conference on Systems, Man and Cybernetics, Vol 4, 3957-3962, (2003)

[22] Situated Interaction Project page: http://research.microsoft.com/en-us/um/people/dbohus/research_situated_interaction.html

[23] K. R. Thórisson. A Mind Model for Multimodal Communicative Creatures and Humanoids, in International Journal of Applied Artificial Intelligence, 13(4-5): 449-486 (1999)

[24] K. R. Thórisson. Natural Turn-Taking Needs No Manual: Computational Theory and Model, From Perception to Action, in Multimodality in Language and Speech Systems, 173-207, Kluwer Academic Publishers (2003)

[25] D. Traum and J. Rickel. Embodied Agents for Multi-party Dialogue, in Immersive Virtual Worlds, AAMAS'02, pp 766-773 (2002)

[26] V-Lingo Mobile - http://www.vlingomobile.com/downloads.html

A 3D Gesture Recognition System for Multimodal Dialog Systems

Robert Neßelrath and Jan Alexandersson

DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
{rnessel,janal}@dfki.de

Abstract

We present a framework for integrating dynamic gestures as a new input modality into arbitrary applications. The framework allows training new gestures and recognizing them as user input with the help of machine learning algorithms. The precision of the gesture recognition is evaluated with special attention to the elderly. We show how this functionality is implemented into our dialogue system and present an example application which allows the system to learn and recognize gestures in a speech based dialogue.

Introduction

Within the last years the number of new technologies and appliances for home entertainment and household has vastly increased, confronting users with a high amount of new electrical devices and heterogeneous interaction concepts. In addition to the often difficult interaction with modern devices like computers and consumer electronics, the interaction with more traditional devices such as television, telephone, and in-car-appliances, gets more and more complicated with the growing number of functionalities. This surplus creates the problem of accessing all of these functionalities in an easy way. Most systems try to solve this problem with a more or less well structured menu concept. In practice, however, a lot of users still have difficulties to navigate through different menus and find the operations they are looking for.

While users with technical expertise and a great interest in modern devices cope with the interaction after some time, especially the elderly or persons with cognitive disabilities never get a chance to use all electrical devices in their home. Within the i2home project [1] new concepts for ambient assisted living are developed based on existing and evolving industry standards, for example the Universal Remote Console (URC) Standard [2]. One approach [3] is based on the multimodal dialogue platform ODP which has been used successfully in several research and industrial projects [4]. Our dialogue system supports the user in the problem-solving process by taking into account the discourse context and providing several communication modalities. Input commands of the user are given via speech, click gestures, or a combination of these. System answers are given either visually or acoustically through speech and sounds. The i2home dialog system allows the user to control its environment by speech interaction and a well designed and easy to use graphical user interface. This allows user to control their kitchen, a reminder and television.

In the future we will develop systems that address the needs of single users as well as user groups not only with cognitive but also with physical disabilities. That confronts us with the individual problems of users that are too limited to interact with a system even by click gestures or speech. In order to provide the opportunity for those persons to communicate with the system, new input devices and modalities must be included. These could be gyroscopes and accelerators but also eye-trackers. All of these devices have the advantage that they not depend on humans using their fingers to manipulate their environment, as they do traditionally. This gives us the opportunity to enhance our input modalities by taking into account other aspects like movements of arms, eyes or head.

In this paper we introduce the integration of dynamic gesture input into the multimodal dialogue system. To record the gestures we use an accelerometer which is integrated in the common input device for the Nintendo Wii, the Wii Remote. The measured values are used to describe the movement of the arm in a three dimensional space and are trained with machine learning systems, in order to recognize the executed dynamic gestures.

The next chapter introduces gestures as part of human communication and demonstrates their use in man-machine communication. The following one deals with TaKG, a toolkit for classifying gestures, and its integration into our dialogue system. For usability tests we evaluated the gesture recognition system with elderly persons in a retirement home.

Gestures

Human communication is a combination of speech and gestures. Gestures are part of the nonverbal conversation and are used consciously as well as subconsciously. Gestures are a basic concept of communication and were used by humans even before speech developed, they have the potential to be a huge enrichment to an intuitive man-machine communication.

One distinguishes between dynamic and static gestures. Static gestures are used for finger spelling among other things. In this case only the position of hand and the alignment of the fingers provide the information for the communicative act. Dynamic gestures additionally contain a movement and in most cases have either a pantomimic meaning, i.e. imitating an action or a symbolic meaning, for example waving to someone.

Gestures in Related Projects

Generally, there are two ways for recording gestures. Non-instrumental projects recognize hand and finger postures with cameras and image processing algorithms [5]. Other projects use instruments for recording, for example sensor gloves or hand devices with integrated sensors like accelerometers or gyroscopes [6] [7]. This is also the concept of the Wii game console and it is their device that is used for this project. In Wii games, often easy properties are used for interpreting gestures, for example the strength and the direction of a movement. The tool LiveMotion[1] from Ai-Live is a framework for Wii game developers focused on learning and recognizing more complex gestures. The creation of motion recognizers is mastered by showing gesture examples without coding or scripting. Recognition should be very fast and without using buttons but is only usable by game developers who have a contract with Nintendo.

A worthwhile goal to use gestures as input is to integrate sensors into devices of everyday life and to recognize device-related gestures. For example the gestures used during operating a mobile device can be taken to recognize scenarios, for example picking up a ringing mobile phone from the table and hold it to the ear as a scenario for accepting a call [8].

Wii Remote Acceleration Sensors

In this work the movement of a dynamic gesture is detected by an ADXL330 accelerometer which is integrated in the Wii Remote controller. The ADXL330 measures acceleration values with 3 axis sensing in the interval -/+ 3g. The acceleration is described in a right-handed Cartesian coordinate system. A Wii Remote, which lies bottom side down on a table, measures the value of 1g in the direction of the z-axis. This is the force the hand needs to exert against gravity and thus an unmoved Wii Remote always measures the absolute acceleration value of 1g. In free fall the absolute value is zero.
The complete movement of the hand within the three-dimensional space can be described by observing acceleration in a series respective to the time. Figure 1 shows the x-axis measurement of a hand movement to the left, that is the axis of interest for this movement. First the curve contains positive amplitude for the acceleration. During the time of the movement with constant speed the acceleration value first goes down to zero and then deceleration causes it to become negative. Figure 2 shows the measurement of a movement to the right and the same curve is observed only with amplitudes in opposite directions. Since hand gestures take place in 3D-space, the sensor values of three accelerometers are part of one measurement. Figure 3 shows the measurement of a circle movement.

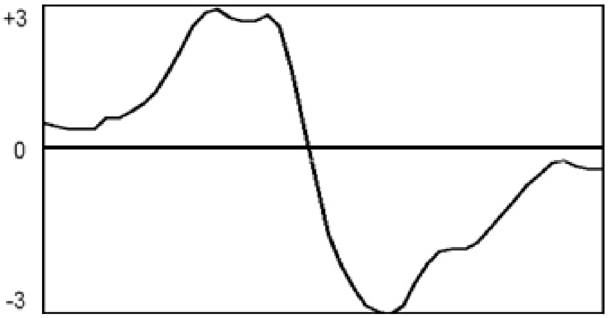


Figure 1: Acceleration data of a movement to the left

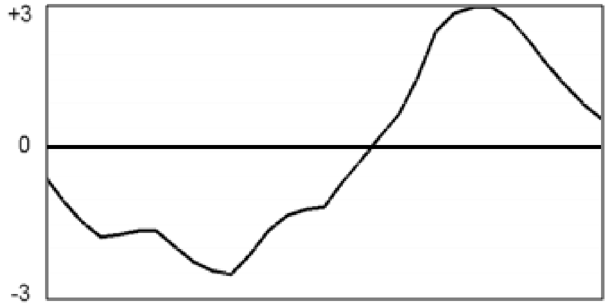


Figure 2: Acceleration data of a movement to the right

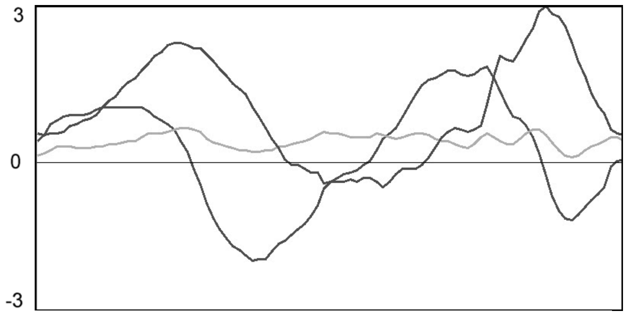


Figure 3: 3D- acceleration data of a circle gesture

WiiMote Recorded Gestures

The character of the WiiMote as an input device leads to some technical constraints which influence the gesture type we can recognize. This includes only dynamic gestures, i.e. the movement of the hand, which is described and modified by the movement path relative to the start position, as well as by the alignment of the WiiMote holding hand, and by the movement speed.

In difference to mouse gesture recognition frameworks [9] or gesture controlled internet browsers [10], the recogni-

[1] http://www.ailive.net/

tion of WiiMote recorded gestures is not limited to two dimensions. Movement in space includes the third dimension and leads to different problems than the recognition of mouse gestures in 2D-space. The 3D-movements are described by their acceleration values, mouse gestures by an array of positions on a plane.

Gesture Recognition

The path of a gesture recorded by the WiiMote involves all three dimensions, giving multidimensional time series in which not only exceptional measurements are included but also their temporal relations to other dimensions. For gesture recognition this means that it is not sufficient to examine the dimensions separately. There must also be synchronization between them.

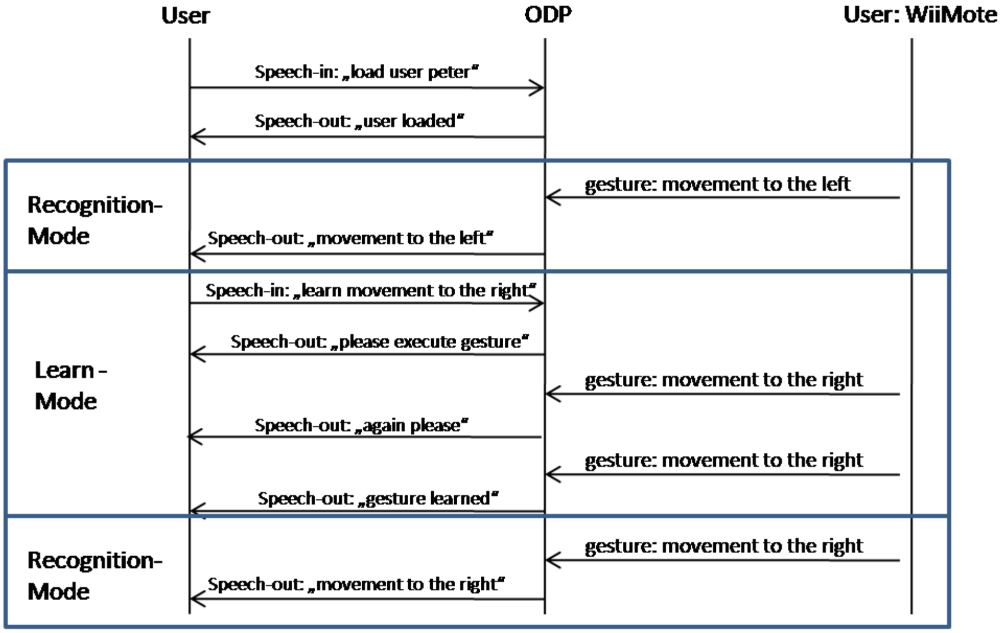

User
ODP
User: WiiMote
Speech-in: „load user peter"
Speech-out: „user loaded"
Recognition-Mode
gesture: movement to the left
Speech-out: „movement to the left"
Learn - Mode
Speech-in: „learn movement to the right"
Speech-out: „please execute gesture"
gesture: movement to the right
Speech-out: „again please"
gesture: movement to the right
Speech-out: „gesture learned"
Recognition-Mode
gesture: movement to the right
Speech-out: „movement to the right"

Figure 4: Typical cycle of the gesture training and recognition dialog

Another problem is that the measurements for only one gesture differ in time, movement-path and speed with every execution. Comparison of two measurements of the same gesture thus has to handle warps in a non-linear way by shrinking or expanding along the time axis and also the single space axes.

An algorithm which is often used to measure similarities between two signals is the Dynamic Time Warping (DTW). This algorithm calculates the distance between each possible pair of points out of two signals and finds the least expensive path through the resulting distance matrix using dynamic programming. The resulting path expresses the ideal warp between the two signals and synchronizes the signal in order to minimize the distance between the synchronized points. With some adaption it can also be used for multi-dimensional gestures [11].

Another approach is to use machine learning algorithms for classifying data. The WEKA framework [12] is a collection of machine learning algorithms implemented in Java and provides interfaces for the easy usage of the most common algorithms. Besides the DTW we also test algorithms that are included in WEKA to learn and recognize gestures: Support Vector Machines (SVM) und Neuronal Networks (NN). Because these algorithms need a fix number of attributes for an instance the acceleration data is preprocessed for input. This includes normalization of the values and interpolation of the measurement on a fix sized set of sampling points.

The TaKG Framework

The framework TaKG [13] is a toolkit for gesture recognition and serves to simplify the integration of gesture controlled interaction into applications. It implements needed functionalities for signal feature extraction and the recognition algorithms mentioned in the chapter before: SVM, NN and DTW.

Furthermore, TaKG is responsible for learning new gestures and organizing them into user specific training sets. Within a set, the information for every trained gesture is listed including the measured signal data and a *gesture tag* denoting the gesture. The main API contains the following functionalities: Load data for a special user, learn and delete gestures and classify new recorded gestures.

A gesture classifying request returns the gesture tag of the gesture in the training set with the highest similarity to the gesture which has been provided together with the request. Another option is to ask for a ranked list of all trained gestures. SVM and NN provide just a ranking, the DTW algorithm describes similarity based on Euclidian distance.

Gesture controlled calculator

One example application which was also used for evaluating the gesture recognition precision is a simplified calculator. The calculator contains buttons for the digits from 0-9, the operators plus (+), minus (-) and equals (=) and a clear button. Every button can be pressed by painting the appropriate figure into the air (or an arbitrary gesture the user associates with the button). The movement depends on the gestures the user performed for training the gesture classifier.

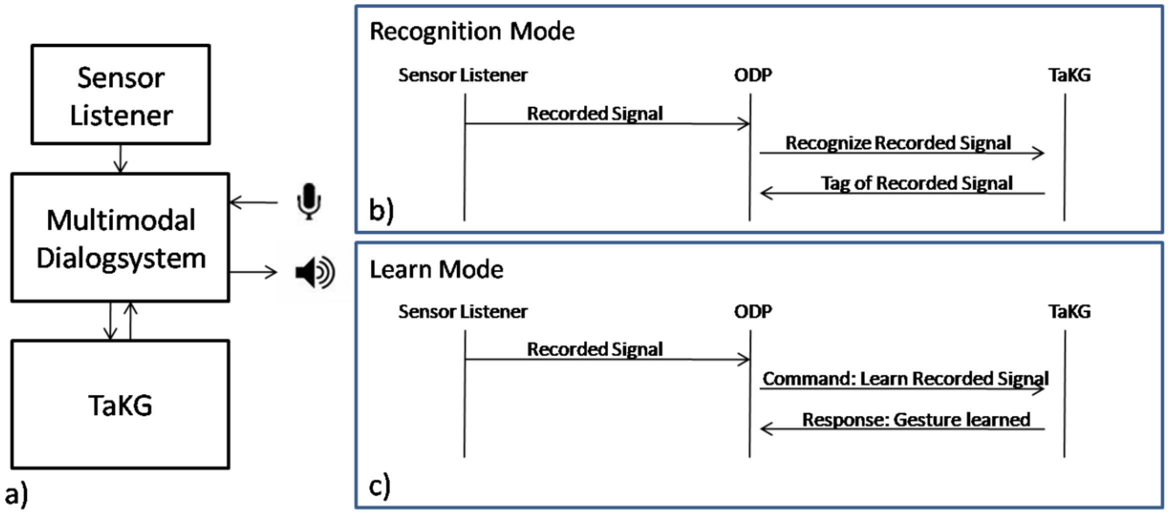


Figure 5: a) Module overview of the gesture training and recognition dialogue. b) Message Flow in Recognition Mode c) Message Flow in Learn Mode

Integration into a dialog system

Gesture Training and Recognition Dialog

The main intention in building a gesture recognition system was to integrate it into our multimodal dialogue system. Our example dialogue application uses speech and gestures to train the classifier with new gestures. During this process we distinguish between two signal processing modes, one mode is the learning mode, the other one the recognizing mode. The standard mode is ‘recognize’, this means that every incoming signal is classified and the detected tag is synthesized for speech output. By voice command the user can set the system into learn mode which adds a new received signal to the training data set. Figure 3 shows a typical dialogue between a user and the system. Additional speech commands allow the user to manage the training data set, i.e. to load data sets for other users or to delete already learned gestures or even complete gesture information about a user.

For implementation of gesture recognition the dialogue system is expanded with two new modules (fig. 4a). One module, the sensor listener, is informed about new recorded gestures. The second module addresses the classifier and works as an adapter to the TaKG.

The dialogue system in the middle is responsible for context-based reaction on new sensor inputs and the communication with the user. In recognition mode new recorded signals are sent to TaKG with a recognition request (fig. 4b). The received tag for the gestures is send to speech-synthesis and the user gets a speech reaction, announcing the user the recognized gesture. In learn mode, ODP sends a learn request to TaKG and controls whether the gesture has been learnt successfully. The result is committed to the user via speech.

Evaluation

The gesture system was evaluated with a heterogeneous group of participants, differing in age and gender. Thus we avoided that only younger persons with experiences in using modern input devices attended the test. Since we were especially interested in how elderly people would respond

Table 1: Evaluation Results. The numbers present the percentage of the correct recognized gestures.

Proband-Id	SVM	NN	DTW
Age 20-50			
M1	76 %	81 %	81 %
W1	**93** %	**93** %	**93** %
W2	76 %	64 %	55 %
M3	88 %	90 %	62 %
M4	79 %	69 %	69 %
M5	74 %	62 %	40 %
W3	83 %	83 %	79 %
Mean	**81** %	77 %	68 %
Age 50-90			
W4	83 %	81 %	**86** %
W5	60 %	57 %	40 %
W6	48 %	55 %	57 %
W9	55 %	55 %	50 %
Mean	**62** %	**62** %	58 %
Age 90+			
W7	**71** %	67 %	69 %
W8	45 %	38 %	43 %
W10	57 %	43 %	48 %
Mean	**58** %	49 %	53 %
General Average	**66** %	63 %	60 %

to the system, we conducted some tests in a retirement home. The following participants took part in the testing:

- 7 persons aged between 20 and 50 years
- 4 persons from 50-90 years
- 3 persons older than 90 years

A fourth person in the 90+ group decided not to participate.

The evaluation helped to answer several questions. Our first interest was to analyze the recognition quality of the gesture recognition algorithm. For this we evaluated the recorded gesture information with all the implemented gesture recognition algorithms.

Furthermore, we observed how users from different age groups dealt with gesture control. Certainly the number of participants is too low to assure statistical significance but we get a first insight how even elderly handle gesture controlled applications.

Test scenario

For the test scenario we used the previously mentioned gesture controlled calculator. Every participant first trained the system with his own gestures for the different digits and operators. For this every gesture was recorded three times. Most of the participants used figures that were similar to that of drawing the number/operator on the black board.

After the system was trained the participants had to solve three different arithmetic problems which were read out loud by the test leader. The users were not informed whether or not a gesture was correctly recognized, in order to avoid that this would have an effect on how they realized the specific gestures. All gestures were recorded and later evaluated with the different recognition methods.

Recognition results

Table 1 shows an overview over the evaluation results. Support Vector Machines (SVM), Neuronal Networks (NN) and the Dynamic Time Warp algorithms (DTW) were used for the precision tests. We observe that the modern machine learning algorithms have an advantage over the dynamic time warp. Results were especially striking for younger people who reach an average precision of 81 % while people who are ninety years and above still achieve an accuracy of more than half of the gestures being recognized correctly. When examining this result we should take into account that the participants only had a relatively short training phase to get used to the gesture interaction. We suspect that after a longer learning phase, the performance of the gestures and thus the precision would improve for this group as well.

A closer look to the confusion matrix in figure 5 reveals that the most of the mistakes were made with gestures which are very similar in their movement paths. For example, mistakes often occurred between zero and six (0--6) or one and seven (1—7). For practical use this can be avoided by defining sufficiently different gestures. One example for this is the pre-given alphabet of handwriting recognition on mobile devices with touchscreen.

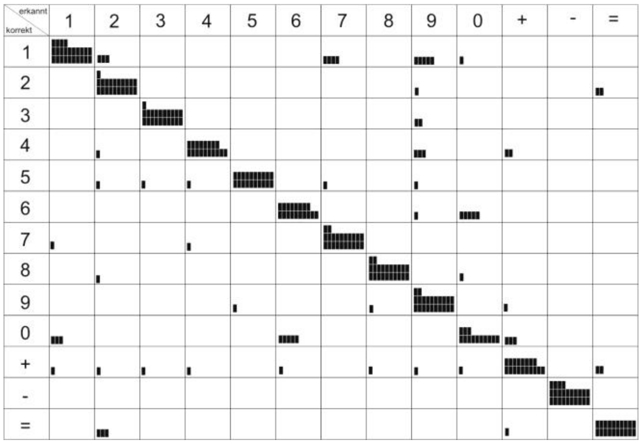


Figure 6: Confusion matrix. Lines contain the executed gestures, columns the recognized. Every rectangle is one gesture input.

Conclusion & Outlook

We introduced TaKG, a toolkit for automatic classification of gestures, and showed how it is integrated as a new input modality into a multimodal dialogue system implemented using the ODP framework. The classifier can be trained with tagged 3D-gestures which are recorded by a WiiMote. New gestures are classified by their gesture tag, which serves as input for the dialogue system and can be processed with other input modalities. An example application shows how to use gesture recognition in combination with speech synthesis. A small spoken dialogue model is used to guide the user for training the classifier and then switch to recognition mode.

Evaluation showed that even elderly persons are able to effectively use the WiiMote as an input modality and that the recognition results are very precise, although the participants did not have a very long training phase.

In the future, we will integrate the gesture modality into system aiming at scenarios in everyday life. Relevant applications include consumer electronic equipment, e.g. TV or media player. Here, a quick move to the right could switch to the next song or channel, a move to the left to the previous one. A direct channel access could be realized by writing the number of the channel into the air like in the calculator example. Turning the hand influences the volume and a fast slash could mute the sound. Since the system allow the user to train the system with his/her own gestures, new gestures can be introduced for specific music genres etc. Performing a gesture would create a play list which only contains songs of the genre the user related to his gesture. Furthermore we want to combine deictic and symbolic gestures. For this we take advantage of the Wii-

Mote IR sensors. This allows us perform gestures relative to an object presented on a monitor or objects in a room.

A further research interest is to move away from the Wii-Mote and to use the signal classifier for other input devices with different sensors. A follow up project deals with a highly personalized dialogue system, especially for disabled persons. Here it is important to support various different devices and sensors which are adapted to the abilities of a single person. The gesture classification algorithms introduced in this paper are independent from a special device and can be used to indentify signals from diverse sensors, giving the dialogue system flexibility in its input modalities.

Literature

[1] *Intuitive Interaction for Everyone with Home Appliances based on Industry Standards* (i2home). http://www.i2home.org/

[2] J. Alexandersson, G. Zimmermann, J. Bund. *User Interfaces for AAL: How Can I Satisfy All Users?*. In Proceedings of Ambient Assisted Living - AAL. 2. Deutscher Kongress mit Ausstellung/Technologien - Anwendungen - Management, pp. 5-9, Berlin, Germany, 2009.

[3] A. Pfalzgraf, N. Pfleger, J. Schehl, J. Steigner. *ODP - Ontology-based Dialogue Platform*. Technical Report, 2008, SemVox GmbH.
http://www.semvox.de/ whitepapers/odp_whitepaper.pdf

[4] J. Schehl, A. Pfalzgraf, N. Pfleger, J. Steigner. *The BabbleTunes System - Talk to your iPod!* To appear in: Proc. of the 10th International Conference on Multimodal Interfaces (ICMI 2008), pp 77-80, Crete, Greece.

[5] I. Laptev, T. Lindeberg. *Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features*. Technical report, 2001, Department of Numerical Analysis and Computer Science, KTH (Royal Institute of Technology), Stockholm, Sweden.

[6] D. Wilsona, A.Wilson. *Gesture Recognition Using the Xwand*. Technical report, Assistive Intelligent Environments Group Robotics Institute Carnegie Mellon University and Microsoft Research.
http://www.cs.cmu.edu/~dwilson/papers.xwand.pdf

[7] A. Y. Bensabat, J.A. Paradiso, *An Inertial Measurement Framework for Gesture Recognition and Applications*, 2001, In Gesture Workshop, pp 9-20.

[8] V. M. Mäntylä, J. Mäntyjärvi, T. Seppänen. *Hand Gesture Recognition of a Mobile Device User*. In Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on, pp 281-284.

[9] B. Signer, U. Kurmann, M. C. Norrie. *iGesture: A General Gesture Recognition Framework.* In Proceedings of ICDAR 2007, 9th International Conference on Document Analysis and Recognition, 954-958, Curitiba, Sept.2007.

[10] Mouse Gesture Plugin*, Optimoz Team*
http://optimoz.mozdev.org/gestures/index.html

[11] G.A. ten Holt, M.J.T. Reinders, E.A. Hendriks. *Multi-Dimensional Dynamic Time Warping for Gesture Recognition*. Thirteenth annual conference of the Advanced School for Computing and Imaging, 2007.

[12] G. Holmes, A. Donkin, I. H. Witten. *Weka: A Machine Learning Workbench*, Technical report, Department of Computer Science, University of Waikato, 1994.

[13] R. Neßelrath. *TaKG: A toolkit for automatic classification of gestures*, Masterthesis, Saarland University, 2008

Dialog Modeling Within Intelligent Agent Modeling

Marjorie McShane and Sergei Nirenburg

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21136
{marge,sergei}@umbc.edu

Abstract

In this paper we explore to what extent the modeling of dialog can be subsumed under the broader modeling of intelligent agents. Specifically, we analyze how four core aspects of modeling human physiology and general cognition can be applied to dialog, and attempt to determine whether dialog poses any modeling challenges that require additional methods or expressive power.

Introduction

Novel applications can challenge traditional conceptions of how best to tackle a problem. Sometimes this happens more than once. A case in point: in the late 1970s Perrault and Allen (see, e.g., Allen and Perrault 1980) suggested a goal- and plan-based approach to dialog processing, influenced by classical AI approaches to planning and the work of Cohen, Levesque and Perrault (e.g., Cohen 1978, Cohen and Perrault 1979, Cohen and Levesque 1980). Later work in dialog processing (cf., e.g., Larsson et al. 2002, Lemon and Gruenstein 2004) has come to rely predominantly on dialog cues. This latter approach has moved away from (a) integration with a general goal- and plan-based conception of agency and (b) applying goals and plans specifically to dialog processing. This choice of this approach was justified in terms of feasibility understood in at least two different but complementary ways. The first – pessimism with respect to massive knowledge acquisition – has quickly propagated from its origins in expert systems to natural language processing and has strongly contributed to the paradigmatic shift toward knowledge-lean methodologies, stochastic ones chief among them. Second, building comprehensive models of intelligent agents was considered to be beyond the scope of a single research team; therefore, general problem solving and dialog processing were adjudged to have a better chance of success through the collaboration of teams working on these two issues.

As a result of these two feasibility-oriented concerns, a cornerstone of dialog processing has been its reliance on dialog models crafted for, and solely dedicated to, dialog. This is true even of DIPPER (Bos et al. 2003), a dialog system architecture that (a) claims to combine the strengths of the goal-oriented and cue-oriented paradigms implementing the information-state approach (Traum and Andersen 1999) and (b) uses "aspects of dialogue state as well as the potential to include detailed semantic representations and notions of obligation, commitment, beliefs and plans" (Bos et al. 2003).

From a purely scientific point of view, participation in dialog is just one of many capabilities inherent in a cognitively complex intelligent agent. It is natural, therefore, that with the growing prominence of work on naturally inspired intelligent agents that emulate human performance over a broad spectrum of everyday and specialized tasks, the separation of dialog processing from general problem solving capabilities has started to look increasingly less justified. Indeed, even from the standpoint of efficiency, integrating the two functionalities promises to bring about the benefits of economies of scale.

We have been exploring issues related to incorporating dialog into a comprehensive intelligent agent at our current stage of work on Maryland Virtual Patient (MVP)[1], an agent-oriented simulation and mentoring environment aimed at automating certain facets of medical education and assessment. The agent network in the system is composed of both human and artificial agents. The human agents include the user (typically, a trainee) discharging the duties of an attending physician and, optionally, a human mentor. The software agents that are built to simulate human behavior include the virtual patient, lab technicians, specialist consultants and a mentoring agent. The system also includes an array of non-humanlike software agents.

The core agent is the virtual patient (VP), a knowledge-based model and simulation of a person suffering from one or more diseases (e.g., Jarrell et al. 2007, 2008; McShane et al. 2007a,b). The virtual patient is a "double agent" in that it models and simulates both the physiological and the cognitive functionality of a human. Physiologically, it undergoes both normal and pathological processes in re-

[1] Patent pending.

sponse to internal and external stimuli. Cognitively, it experiences symptoms, has medical and lifestyle preferences (a model of character traits), is capable of remembering, forgetting and learning, makes decisions, and communicates with the trainee about its personal history, symptoms and preferences for treatment. Users can interview a VP; order lab tests; receive the results of lab tests from technician agents; receive interpretations of lab tests from consulting physician agents; posit hypotheses, clinical diagnoses and definitive diagnoses; prescribe treatments; follow-up after those treatments to judge their efficacy; follow a patient's condition over an extended period of time; and, if desired, receive mentoring from the automatic mentor.

Work on MVP has progressed from the simpler (although not simple!) development of realistic, interactive physiological simulations of a half dozen diseases to the development of the cognitive functioning of VPs, which includes interoception (the experiencing of internally generated stimuli, like symptoms), decision-making, memory management, natural language processing, and the ability to learn lexical and ontological information. All of these capabilities have been modeled and implemented in a demonstration version of the system. The current system does not, however, include a dedicated dialog model. As we progress toward enhancing the VP's – as well as the mentor's – language capabilities, we are reviewing what have been considered to be central components of dialog models and the extent to which they can or cannot be subsumed under our generalized cognitive architecture. In other words, can we generalize without losing precision and compromising on depth of coverage? In this paper, we examine some theoretical and practical issues involved in making this judgment, focusing on four aspects of cognitive modeling that have application both inside and outside of the realm of dialog: (1) meaning representation, (2) the use of remembered goals, plans and scripts, (3) decision-making, and (4) learning.

Meaning Representation

The representation of meaning is central to semantically-oriented text processing. When working with meaning, many NLP practitioners find it appropriate to use elements of a metalanguage and surface strings in conjunction: e.g., case roles (e.g., Gildea and Jurafsky 2002) or template slots (e.g., Hobbs et al. 1997)) might be filled by strings. Others consider it necessary to represent meaning using a metalanguage independent of any natural language. This requires that language strings be interpreted and converted into and out of that metalanguage. While the latter approach is typically considered more expensive, since semantic analysis and the component disambiguation are notoriously difficult, it offers clear benefits to an NLP system, permitting it to do its main work over unambiguous meaning representations. (See Bar Hillel 1970 for a discussion of the division between NLP and reasoning.) Moreover, if an intelligent agent has functionalities beyond language processing (as is assumed, e.g., also in Allen et al. 2001), the use of the same metalanguage for the representation of both linguistic and non-linguistic meaning offers even greater benefits, as evidenced by our experience with MVP.

In MVP, all physiological, general cognitive and language processing capabilities of all agents rely on the same ontological substrate, the same organization of the fact repository (agent memory of assertions) and the same approach to knowledge representation. This approach, which was originally developed for the knowledge-based semantic processing of language – and, in fact, has been used in the OntoSem semantic analyzer and its predecessors for almost two decades (see, e.g., Nirenburg and Raskin 2004; Beale et al. 1997) – has been seamlessly applied to the representation of all meaning in MVP, from physiological simulation to all aspects of agent knowledge, memory and reasoning. So, we seem to have a constructive proof that not only is a special type of meaning representation for dialog not necessary, it would be detrimental to overall agent functionality since it would require the agent to treat information from different sources – in MVP, interoception, language input and the results of the agent's own reasoning – in different ways, which is neither efficient nor conceptually plausible. It is noteworthy that basically no extensions to the expressive power of the OntoSem ontology or metalanguage have been required to support this new application: indeed, there is no fundamental difference between a script that permits a language processing agent to interpret a restaurant scenario (to choose an example at random) and a script describing the progression of a disease that permits the operation of a simulation. Although space does not permit a detailed discussion of the OntoSem and MVP environments, select observations and examples should make clear the benefits of a unified approach to meaning representation for a multi-functional intelligent agent.

Our meaning representations (MRs) are composed of ontological concepts and their instances, linked by ontologically defined properties, using the OntoSem ontology as the substrate. The interpretation and generation of language invokes an English lexicon whose entries are described with reference to ontological concepts, either directly or with property-based modifications (McShane et al. 2005a). When MRs convey the meaning of text we call them TMRs (text meaning representations) – a topic we have written about extensively in the past. Consider the following example, which shows various aspects of MR creation and use in MVP:

1. The physiological disease simulation creates an instance of pain in the VP at a certain time and with a certain intensity. Let us call this instance, which is remembered by the system as a trace, MR-1. This pain is experienced via (simulated) interoception by the VP. The VP's MR, MR-2, may or may not look like MR-1 due to the fact that the VP's ontology is not expected to match that of the "expert" ontology that underpins the simulation. For example, the

simulation might invoke ontological concepts (e.g., PERISTALSIS) that are known to a physician but not to a lay person. As a result of this mismatch, the system must carry out ontological paraphrase to support VP interoception, permitting the VP to interpret and remember the symptom in its own terms (for further discussion, see McShane et al. 2008a,b).

2. If the user asks the VP about pain, the VP interprets the English string, creating a TMR – we'll call it MR-3 – whose elements ontologically match those of the VP's memory. (For paraphrase in this context, see McShane 2008a,b.) The VP records MR-3 in its memory, as part of its dialog record, and uses its content to look up the appropriate response in its memory.

3. When the VP has found the answer, it generates a TMR, MR-4, that acts as input to the English language generator. In other words, this TMR constitutes the result of the content specification step in generation.

4. When the automatic mentor observes and records the user-VP conversation, it uses the MRs generated and interpreted by the VP, not needing to independently process English strings. By contrast, when the mentor converses with the user, it carries out the same kind of language processing and memory management as does the VP.

In sum, all of the knowledge stored and manipulated in MVP is represented using the same metalanguage, with conversion to and from English needed only for VP-user interactions and mentor-user interactions.

As an aside, there actually is one other use for metalanguage-to-English conversion in MVP: it is used to show the VP's thinking processes as part of the under-the-hood view of the simulation, where the knowledge structures being produced throughout the simulation can be viewed dynamically. Generating an English "window" into the workings of the system is, of course, not a native part of the system; however, this is useful when explaining the system to observers, since the formal metalanguage structures cannot be easily read and interpreted. To take one example: say the VP has been seen by the doctor, who may or may not intervene, and has been told to come back for a follow-up in 9 months. However, at 7 months the VP experiences a spike in symptoms. Whether or not the VP will decide to present early – and if so, when it will present – depends, in the current implementation, upon the actual level of the symptom(s), the VP's symptom threshold (how severe symptoms have to be in order to take action), the amount of time left before the scheduled appointment, and the degree of the sudden change in symptoms. (Of course, one could add any number of other factors, such as how much the VP likes or dislikes going to the doctor, whether the VP has time to see the doctor, how nervous the VP is about its health, etc. These are planned for the future.) At a given point in time the VP's thoughts might be: *My symptoms have increased significantly. They were mild when I saw the doctor last and now they are moderate. I have a high tolerance for symptoms. My appointment is 2 months away. I will wait until my appointment.* Of course, this type of evaluation is carried out regularly, so the VP might still decide to present early. Each of these "thoughts" is automatically generated from the functions that comprise the VP's decision-making process.

Scripts, Plans and Goals

Dialog models must include mechanisms that permit the conversational agent to know what to do next – if a question is posed the agent should answer it, if the interlocutor clearly didn't understand an utterance, the agent should clarify (this is an aspect of "grounding", as discussed, e.g., in Traum 1994), and so on. Allen and Perrault (1980) introduced goal- and plan-based reasoning in the study of dialog interpretation. Later frameworks preferred to rely on the notion of discourse obligation (e.g., Traum and Allen 1994). That is, the interlocutor's utterance sets up a discourse obligation on the part of the agent, and the agent must fulfill this obligation in the way specified by an associated function. The hypothesis we have been investigating is that agent behavior in a dialog can be modeled using the same goal-driven methods used to initiate all other agent action in MVP, without the need for dialog-specific obligations. To show why we think this is possible and preferable – at least for our multi-faceted agents – we must start with an overview of the use of goals, plans and scripts in MVP.

As mentioned earlier, the MVP simulation employs ontologically recorded complex chains of events. We distinguish two types based on agency and the extent to which the events are goal-driven.

Scripts in our approach are unagentive complex events for which positing goals would be a stretch, since the goals would need to be attributed to non-sentient, questionably sentient (e.g., bacteria) or divine sources: a heart beats (in order to keep the human in which it resides alive); a disease progresses (for whatever goals the instruments of disease – e.g., bacteria – fulfill by perpetuating the disease); food passes from the esophagus to the stomach (so that it can nourish the eater). Let us consider the example of a disease script more closely. In MVP, diseases are modeled as processes (non-humanlike agents) that cause changes in key property values of a VP over time. For each disease, a set number of conceptual stages is established and typical values or ranges of values for each property are associated with each stage. Relevant property values at the start and end of each stage are recorded explicitly, while values for times between stage boundaries are interpolated. The interpolation currently uses a linear function, though other functions could as easily be employed. A disease model includes a combination of fixed and variable features. For example, although the number of conceptual stages for a given disease is fixed, the duration of each stage is variable. Similarly, although the values for some physiological properties undergo fixed changes across patients, the values for other physiological properties are variable across patients, within a specified range. The combination of fixed and variable features represents, we believe, the golden mean for disease modeling. On the one hand, each disease model is sufficiently constrained so that patients suffering from the disease show appropriate physiological manifestations of it. On the other hand, each disease model is sufficiently flexible to permit individual patients to differ in clinically relevant ways, as selected by patient authors. (See Jarrell 2007, 2008 and McShane 2007a,b for detailed descriptions of disease models.)

Plans, by contrast, are agentive and are used as a means of satisfying some agent's **goal**: going to the doctor and buying over-the-counter medication are two plans for the goal of healing illness; eating and taking diet pills are two plans for the goal of satisfying hunger; accepting the doctor's recommendation for a medical test and inquiring about other diagnostic options are two plans for fulfilling the goal of diagnosing disease. Goals are states, and states are formally represented in the OntoSem framework as objects with specific property values. For example, the goal of being healthy, held by a particular person, is represented as (human-x (health-attribute 1)), where 1 signifies the highest value on the abstract scale [0,1].

What follows is an informal sketch of the agent's manipulation of goals and plans in the current version of the MVP environment. The main goal pursued by all VPs in our environment is BE-HEALTHY. We assume that this is a universal goal of all humans and, in cases in which it seems that a person is not fulfilling this goal – e.g., a person makes himself ill in order to be cared for by medical professionals, or a patient selects bad lifestyle habits that damage his health – he is simply prioritizing some other goal, like BE-FOCUS-OF-ATTENTION or EXPERIENCE-PLEASURE, over BE-HEALTHY. In MVP, when a VP begins to detect symptoms, the goal BE-HEALTHY is put on the goal and plan agenda. It remains on the agenda and is re-evaluated when: (a) its intensity or frequency (depending on the symptom) reaches a certain level; (b) a new symptom arises; or (c) a certain amount of time has passed since the patient's last evaluation of its current state of health, given that the patient has an ongoing or recurring symptom or set of symptoms: e.g., "I've had this mild symptom for too long, I should see the doctor." At each evaluation of its state of health, the VP can either do nothing or go to see the doctor – a decision that is made based on an inventory of VP character traits, the current and recent disease state and, if applicable, previous doctor's orders (cf. next section). If it decides to see the doctor, that plan is put on the agenda. All subgoals toward achieving the goal BE-HEALTHY and their associated plans are put on and taken off the agenda based on VP decision functions that are triggered by changes in its physical and mental states throughout the simulation. So when the doctor suggests having a test (goal: HAVE-DIAGNOSIS) and the patient agrees, having the test (a plan toward the above goal) is put on the agenda; and so on.

Returning to dialog modeling, we are attempting to determine whether our plan- and goal-based methods are sufficient to support all the needs of dialog. Although we have not yet explored all of the issues involved, preliminary indications are that they very well may be.

Consider again the use of "obligations" in dialog modeling. Obligations have been used as the explanation for why, e.g., a participant in a dialog must respond to a question even if the answer is essentially vacuous, such as, "I don't wish to respond" (Traum and Allen 1994). However, obligations can be recast in terms of plans and goals: speakers can have the goal of being a polite member of society (which has its own benefits), which in turn has a series of conditional plans: if an interlocutor asks a question, answer it; if the interlocutor has misunderstood, help him or her to understand. These are the same sorts of rules as are found in the obligation-oriented models but their theoretical status is different. The goal BE-POLITE does not exclusively apply to verbal actions in dialogs, it applies as well to physical actions, e.g., not slamming the door in the face of a person entering a building behind you. So, if an intelligent agent – like our VP – is endowed with action capabilities beyond the realm of dialog, generalizations about its overall goals in life should be incorporated rather than splitting up its behavior into dialog-related and non-dialog-related categories.

Another aspect of many dialog models is an agent's understanding of the appropriate interpretation of utterances as dictated by the speech context. For example, when the doctor asks "How are you?" the agent should respond differently than if a colleague had asked the same question. Situation-based disambiguation of this sort can be carried out with the use of ontologically recorded plans. Specifically, an agent's GO-TO-DOCTOR plan encodes its knowledge of what can typically happen at a doctor's visit: greeting, small talk, asking about health and family history, doing a physical exam, positing a hypothesis or diagnosis, discussing treatment options, and so on. Upon receiving a language input, the agent must attempt to match the input to one of the expected points in the plan; if successful, this guides the selection of a response. Use of the same plans can help to detect if the interlocutor has misunderstood an utterance by the agent. In short, any deviation from the expected plan is a clue to the agent that it might need to take repair action (see Traum et al. 1999).

One aspect of dialog modeling that we believe might require a special approach is dialog-specific language conventions, such as: the resolution of personal pronouns; full interpretation of fragments or ellipsis (depending on how one chooses to linguistically analyze structures like "How often?", whose action must be recovered from the previous utterance); semantic ellipsis, which we define as the non-expression of syntactically non-obligatory but semantically obligatory material; etc. (See McShane et al. 2004, 2005b, McShane 2005 for OntoSem approaches to these phenomena.) The approaches to some of these issues will apply to text as well as dialog.

Decision-Making

All cognitive architectures are essentially grounded in a perception – decision – action loop, and the decision-making of all intelligent agents relies on decision functions that take parameter values as input and return a decision as output. Painted in these broad strokes, decision-making in MVP is quite traditional. However, one feature distinguishes the decision-making of VPs from that of most other intelligent agents: for VPs, character traits are central input parameters. The need for distinguishing character traits in creating a large, realistic, highly differentiated population of VPs is undeniable – after all, some patients are fearful, others distrust doctors, still others believe they know more than the physicians they consult… and all of these traits directly impact the patient's behavior in the medical context. However, the utility of modeling character traits is not limited to decision-making about life actions, it extends to the realm of dialog as well: some agents should be talkative and others reticent; some should provide more information than asked for and others should respond in monosyllables; some should use technical terminology and others should use laymen's terms; and so on. Endowing artificial agents with linguistically-oriented traits that permit them to be as distinguished in their mode of communication as they are in other aspects of their lives will, we hypothesize, enhance the suspension of disbelief that is key to an interactive application like MVP.

We have already seen one example of decision-making, using the example of deciding when to present to the doctor. Here we look at a different example in a bit more detail (for more in-depth discussion of decision-making in MVP, see Nirenburg et al. 2008b). Among the decisions a patient must make is whether or not to agree to a test or procedure suggested by the doctor, since many interventions involve some degree of pain, risk, side-effects or general unpleasantness. Some patients have such high levels of trust, suggestibility and courage that they will agree to anything the doctor says without question. All other patients must decide if they have sufficient information about the intervention to make a decision and, once they have enough information, they must decide whether they want to (a) accept the doctor's advice, (b) ask about other options, or (c) reject the doctor's advice. A simplified version of the algorithm for making this decision – the actual decision tree is too detailed to be included here – is as follows.

```
1. IF a function of the patient's trust, suggestibility and courage is above a threshold OR the risk associated with the intervention is below a threshold (e.g., in the case of a blood test)
   THEN it agrees to intervention right away.

2. ELSE IF the patient feels it knows enough about the risks, side-effects and unpleasantness of the intervention (as a result of evaluating the function enough-info-to-evaluate) AND a call to the function evaluate-intervention establishes that the above risks are acceptable
   THEN the patient agrees to the intervention.

3. ELSE IF the patient feels it knows enough about the risks, side-effects and unpleasantness of the intervention AND a call to the function evaluate-intervention establishes that the above risks are not acceptable
   THEN the patient asks about other options
     IF there are other options
     THEN the physician proposes them and control is switched to Step 2.
     ELSE the patient refuses the intervention.

4. ELSE IF the patient does not feel it knows enough about the intervention (as a result of evaluating the function enough-info-to- evaluate)
   THEN the patient asks for information about the specific properties that interest it, based on its character traits: e.g., a cowardly patient will ask about risks, side effects and unpleasantness, whereas a brave but sickly person might only ask about side effects.
     IF a call to the function evaluate-intervention establishes that the above risks are acceptable
     THEN the patient agrees to the intervention.
     ELSE the patient asks about other options
       IF there are other options
       THEN the physician proposes them and control is switched to Step 2.
         ELSE the patient refuses the intervention.
```

The two decision functions called by this function are presented in Nirenburg et al. 2008b.

Not all human-like agents in MVP need be endowed with character traits. For example, the mentor can be completely devoid of personality, both linguistically and non-linguistically, as long as it effectively assists the user as needed. This does not mean that MVP mentors will be undifferentiated – quite the opposite. We have already implemented mentoring settings that make the mentor provide more or less explanatory information and make it intervene at more or fewer types of junctures. In addition, we plan to add to our current mentoring model additional models that reflect the differing clinical beliefs and preferences of different experts. (It should be noted that much of clinical knowledge is derived from the experience of individual physicians, and that experience can differ greatly across physicians, leading to differing, though potentially coexisting, mental models.) However, these differences lie outside the realm of character traits.

Although we have not yet fully incorporated the influence of character traits on dialog behavior into our nascent dialog processing, a simplified distinction was implemented in an earlier demonstration version of MVP. There, VPs were categorized as medically savvy or medically naïve, with the former providing more information than asked for by the user and the latter providing only what the user explicitly asked for, thus requiring the user to ask follow-up questions.

Learning

Within the field of NLP, machine learning can involve (among many other approaches) learning by reading (e.g., Forbus et al. 2007) and learning by being told. The latter idea ascends to McCarthy 1958 and was developed in systems such as Teiresias (Davis 1982) and Klaus (Haas and Hendrix 1983). Being able to rely on the OntoSem text understander, our VP uses a richer and less constrained channel of communication between the teacher and the learner than earlier systems, though, not surprisingly, the quality of text analysis is far from perfect. Still, the VP can already learn by being told (Nirenburg et al. 2008a), and work is underway (e.g., Nirenburg and Oates 2007) on having it learn by reading. However, its learning does not derive from language alone – the VP can learn from its simulated experiences as well. Whether the learning is based on language or non-linguistic experience, the modeling strategy is the same, as is the effect on agent knowledge.

Three VP knowledge bases are augmented during a simulation: the ontology – information about the world in general, the fact repository ("memory of assertions") – facts about instances of events in the world, and the lexicon – the form and meaning of lexical strings. We have already discussed augmentation of the fact repository; here we briefly describe our approach to learning ontology and lexicon.

The VP can learn ontology and lexicon through discourse with the trainee (learning by being told) or by reading texts, e.g., those found on the web (learning by reading). When the VP encounters a new word, it creates a new lexical entry for it with as much meaning as is immediately available. For example, if the trainee says "Your test results reveal that you have achalasia," and if the VP has never heard of achalasia, it can hypothesize – based on its GO-TO-DOCTOR plan – that achalasia is some sort of a disease (we will not nitpick as to the difference between a disease, disorder, etc.). Thus, the lexical entry "achalasia" will be created, mapped to the ontological concept DISEASE. If the doctor provides more information about the disease, the VP uses these descriptions to fill the associated property slots in its DISEASE concept. Of course, all of the text processing and ontology and lexicon population use the metalanguage referred to above. If we have a curious patient (we do not yet, but we will), then that patient can, between doctor visits, search for information on the web to fill out its ontological specification of its disease or any other poorly understood concepts and return to the doctor/trainee for clarifications, questions, and so on. Thus, dialog-based and reading-based learning can co-occur in the VPs of MVP.

As concerns ontology, apart from learning it through language input, VPs can learn it from direct experience. For example, say a VP agrees to a procedure whose pain level it thinks will be acceptable, but during the procedure the VP realizes that the pain level is far greater than it can tolerate; when proposed that procedure again, the VP can decline based on its revised interpretation of the value of "pain" for the procedure.

Discussion

MVP utilizes knowledge-rich approaches to NLP and agent modeling that were more widely pursued 20 or 30 years ago than they are today. Interest in plan- and goal-based R&D, as well as deep-semantic NLP, dwindled when investigators concluded that they were too labor-intensive to support practical applications. However, these conclusions must be put into perspective, particularly when juxtaposing past efforts with MVP.

First, most of the research on plan- and goal-based reasoning was devoted to creating systems that *developed* plans on the fly. In MVP, by contrast, we imposed the constraint that the system would not be required to develop plans, it would only be required to use preconstructed plans. This simplifying constraint is well-suited to MVP since system users will not be asked to solve never before seen types of cases, and the system itself – in the guise of the virtual mentor – will not be asked to invent novel approaches to patient care or fundamentally creative responses to questions. **Second**, in our environment various types of simplifications are possible with no loss in the quality of the final application. For example, we are not attempting to model every known aspect of human physiology, we are modeling only those that are needed to support a lifelike simulation at a grain-size that fulfills all foreseen teaching goals; we are not planning to supply our mentor with all possible mentoring moves (see, e.g., Evans and Michael 2006), only those sufficient to support the needs of medical students, whose primarily learning through MVP will, we hypothesize, derive from trial and error during practice simulations; and we are not attempting (at least not yet) to configure an open-domain conversational agent but, instead, one that converses well in the more constrained – but certainly not toy – domain of doctor-patient interviews, for which we can realistically develop sufficient ontological and lexical support. **Third**, MVP is a high-end application that requires sophisticated simulation, language processing and generalized reasoning capabilities that are not supported by the types of (primarily stochastic) methods that have been of late attracting the most attention in NLP.

In this paper we have discussed the results of our current, application-driven exploration of the possibility of subsuming a "dialog model" under a more generalized approach to agent modeling. Our goal in creating a unified modeling strategy, with little or no need for highly specialized components, is to create intelligent agents whose functionality can expand in any way – as by the addition of vision or haptics – and into any domain without the need for extensions to the base environment. In addition, we are trying to capture overarching generalizations about agent behavior like the one cited above: a human-like agent should respond to a question in dialog for the same reasons as it does not slam the door in the face of someone entering a building behind it. We believe that such conceptual, modeling and implementational generalizations will lead to the development of agents that will not be disposed of every time a new capability is required of them. This, to our minds, is a pressing goal in building the next generation of intelligent agents.

References

Allen, J., Ferguson, G., and Stent, A. 2001. An Architecture for More Realistic Conversational Systems. In *Proceedings of the Conference on Intelligent User Interfaces*, 1-8. January 14-17, Santa Fe, New Mexico.

Allen, J. F., and Perrault, C. R. 1980. Analyzing Intention in Dialogues. *Artificial Intelligence* 15:3, 143-178.

Bar Hillel, Y. 1970. *Aspects of Language*. Jerusalem: Magnes.

Beale, S., Nirenburg, S., and Mahesh, K. 1995. Semantic Analysis in the Mikrokosmos Machine Translation Project. In *Proceedings of the 2nd Symposium on Natural Language Processing*. Bangkok, Thailand.

Bos, J., Klein, E., Lemon, O., and Oka, T. 2003. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *Proceedings of the 4th SIG-Dial Workshop on Discourse and Dialogue*, Sapporo, Japan.

Cohen, P. 1978. On Knowing What to Say: Planning speech acts. Technical Report 118, Department of Computer Science, University of Toronto.

Cohen, P. R., and Levesque, H. J. 1980. Speech Acts and the Recognition of Shared Plans. 1980. In *Proceedings of the 3rd Conference of the Canadian Society for Computational Studies of Intelligence,* Victoria, B.C., pp. 263-271.

Cohen, P. and Perrault, C. R. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science,* 3:177-212.

Davis, R. 1982. Teiresias: Applications of Meta-level Knowledge. In Davis, R., and Lenat, D., *Knowledge-Based Systems in Artificial Intelligence*. New York: McGraw-Hill.

Evens, M., and Michael, J. 2006. *One-on-One Tutoring by Humans and Computers*. New Jersey and London: Lawrence Erlbaum and Associates, Publishers.

Forbus, K., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., and Ureel, L. 2007. Integrating Natural Language Knowledge Representation and Reasoning, and Analogical processing to Learn by Reading. In *Proceedings of AAAI-07.*

Gildea, D., and Jurafsky, D. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3): 245-288.

Haas, N., and Hendrix, G. 1983. Learning by Being Told: Acquiring knowledge for information management. In Michalski, R., Carbonell, J., and Mitchell, T. (eds.), *Machine Learning: An AI Approach*. Palo Alto, CA: Tioga.

Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. 1997. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. *Finite-State Language Processing*, MIT Press, Cambridge, MA.

Jarrell, B., Nirenburg, S., McShane, M., Fantry, G., Beale, S., Mallott, D., and Razcek, J. 2007. An Interactive, Cognitive Simulation of Gastroesophageal Reflux Disease. *Studies in Health Technology Informatics,* 194 –199. IOS Press.

Jarrell, B., Nirenburg, S., McShane, M., Fantry, G., and Beale, S. 2008. Revealing the Conceptual Substrate of Biomedical Cognitive Models to the Wider Community. *Studies in Health Technology Informatics*, 281 – 286. IOS Press.

Larsson, S., Berman, A., Grönqvist, L., and Kronlid, F. 2002. TrindiKit 3.0 Manual. SIRIDUS deliverable D6.4.

Lemon, O., and Gruenstein, A. 2004. Multithreaded Context for Robust Conversational Interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*, 11(3): 241-267.

McCarthy, J. 1958. Programs with Common Sense. In *Proceedings of the Symposium on the Mechanization of Thought Processes*. Teddington, UK.

McShane, M., Beale, S., and Nirenburg, S. 2004. OntoSem Methods for Processing Semantic Ellipsis. In *Proceedings of the HLT/NAACL 2004 Workshop on Computational Lexical Semantics*, Boston, MA.

McShane, M., Nirenburg, S., and Beale, S. 2005a. An NLP Lexicon as a Largely Language Independent Resource. *Machine Translation* 19(2): 139-173.

McShane, M., Nirenburg, S., and Beale, S. 2005b. Semantics-Based Resolution of Fragments and Underspecified Structures. *Traitement Automatique des Langues* 46(1): 163-18

McShane, M., Nirenburg, S., Beale, S., Jarrell, B., and Fantry, G. 2007a. Knowledge-Based Modeling and Simulation of Diseases with Highly Differentiated Clinical Manifestations. In *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME 07)*. Amsterdam, The Netherlands, July 7-11.

McShane, M., Fantry, G., Beale, S., Nirenburg, S., Jarrell, B. 2007b. Disease Interaction in Cognitive Simulations for Medical Training. In *Proceedings of the MODSIM World Conference, Medical Track*, Virginia Beach, Sept. 11-13.

McShane, M., Nirenburg, S., and Beale, S. 2008a. Resolving Paraphrases to Support Modeling Language Perception in an Intelligent Agent. In *Proceedings of the Symposium on Semantics in Systems for Text Processing (STEP 2008)*, Venice, Italy.

McShane, M., Nirenburg, S., and Beale, S. 2008b. Two Kinds of Paraphrase in Modeling Embodied Cognitive Agents. In *Proceedings of the Workshop on Biologically Inspired Cognitive Architectures*, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7-9.

Nirenburg, S., McShane, M., and Beale, S. 2008a. A Simulated Physiological/Cognitive "Double Agent". In *Proceedings of the Workshop on Naturally Inspired Cognitive Architectures*, AAAI 2008 Fall Symposium, Washington, D.C., Nov. 7-9.

Nirenburg, S., McShane, M., Beale, S., and Jarrell, B. 2008b. Adaptivity in a Multi-Agent Clinical Simulation System. In *Proceedings of AKRR'08, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Porvoo, Finland, Sept. 17 – 19.

Nirenburg, S., and Oates, T. 2007. Learning by Reading by Learning to Read. In *Proceedings of the International Conference on Semantic Computing*. Irvine, CA, August.

Nirenburg, S., and Raskin, V. 2004. *Ontological Semantics*. Cambridge, MA: MIT Press.

Traum, D. 1994. A Computational Theory of Grounding in Natural Language Conversation. PhD thesis, Department of Computer Science, University of Rochester, Rochester, NY.

Traum, D. R., and Andersen, C. F. 1999. Representations of Dialogue State for Domain and Task Independent Meta-Dialogue.

In *Proceedings of the IJCAI'99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Traum, D., Andersen, C., Chong, Y., Josyula, D., Okamoto, Y., Purang, K., O'Donovan-Anderson, M., and Perlis, D. 1999. Representations of Dialogue State for Domain and Task Independent Meta-Dialogue. *Electronic Transactions on Artificial Intelligence* 3(D):125-152.

Traum, D., and Allen, J. 1994. Discourse Obligations in Dialogue Processing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

The Companions: Hybrid-World Approach

Alexiei Dingli, Yorick Wilks, Roberta Catizone and Weiwei Cheng

University of Sheffield, United Kingdom

alexiei.dingli@um.edu.mt, {y.wilks; r.catizone; w.cheng}@dcs.shef.ac.uk

Abstract

The Companions Hybrid-World model defines a new methodology for dialogue. Previous systems were based on a closed world assumption whereby the knowledge of the dialogue system is stored in a knowledgebase and used throughout the conversation. The Hybrid-World model is still based around a knowledgebase (which is used throughout the interaction with the user) but if the user starts talking about a topic which is not found in the knowledgebase, the system is capable of gathering information from the online world. This information is then used to enrich the knowledgebase and forms the basis of a reply to the user. The system uses mainly two approaches; the first approach is to harvest information from known sites using predefined wrappers. The second approach is to seek information with the help of search engines, extract the relevant parts and use those to provide a reply to the user. This approach of using a combination of a close-world model and an open-world model (when needed) proved itself to be very effective when engaging in dialogue with the user since it is not only restricted to the knowledge in the knowledgebase but can technically use all the relevant information available on the web.

Introduction

The COMPANIONS project aims to change the relationship that currently exists between computers and people. The idea is to give machines a more human-like way of interacting. This is not limited to multi-modal additions such as touch and speech but the companion is designed to be a ‘presence’ that stays with the user for a long period of time. Its purpose is to develop a relationship with the person, learn (through conversational interaction) the persons’ preferences and wishes so as to keep them happy and also to build up a narrative of the user’s life.

This paper will provide a description of the current Senior Companion (SC) prototype which is the result of two years of research which is the first half of a funded IST EC IP project (see below) named COMPANIONS[1]. The idea behind it is to act as a support to elderly persons when they venture into the digital world. Each of us has built through the years a myriad of digital records; pictures from cameras, digital or analogue, as well as videos and written records. The recent proliferation of Web 2.0 applications (O’Reilly 2005) has made it easier for people to post personal information on the web through micro-blogs and social networking sites. The problem with this is that we are finding different snippets containing information about a person spread all over the web thus making the personal management of this information increasingly difficult. The SC aims to link the wealth of digital life information and organize it for the user unasked. Through natural conversation, the SC will elicit their life memories by, for example, discussing the personal photographs of the person, using the conversation to get to know information about the user’s likes, dislikes, emotions, etc. Since our approach is multimodal, this helps us recognise the user’s preferences fairly accurately while reducing the uncertainties (Lee, Narayanan, & Pieraccini, 2002, Lauria, 2007). We are further assuming that all of this information will eventually be stored online (as in the Memories for Life project[2]). In essence, the SC aims to create a coherent narrative for its user whilst assisting the person in their daily tasks, while amusing and inforing them when necessary and, most important of all, gaining the trust of the person.

From a technical perspective, the project uses a number of derived machine learning (ML) techniques initially developed in past projects at the University of Sheffield. These techniques combine the use of both a closed world and an open world, to help the SC achieve its objectives. This is why we are calling it a Hybrid-World (HW) approach. The system makes use of a Natural Language Understanding (NLU) module which in turn uses an Information Extraction (IE) approach to derive content from user input utterances of any length. Then it uses a training method for attaching Dialogue Acts to these utterances (Webb, et al.). Using a specific type of Dialogue Manager (DM) that uses a stack and a range of Dialogue Act Forms (DAF) representing particular contexts and meta-dialogue moves, it determines the context of utterance and the stack is used to manage the flow of dialogue. The system uses a mixed initiative approach whereby the initiative can be either of the SC (when it is asking for information about, say, a photo on the screen) or the user (when requesting the system to show, say, all photos of a wedding). Although our current implementation is based upon a standard computer, it can be embodied in different forms such as a mobile device, a screen or even a robot. However, while doing so, the SC will still retain its original personality which is capable of existing on different devices whilst still maintaining a complete dedication to the user. In the following sections, we shall:

1. describe the functionality of the current prototype;
2. sketch its architecture and modules;
3. explain in detail the hybrid-world approach used by the Natural Language Understanding module and the Dialogue Manager.

[1] http://www.companions-project.org/

[2] http://www.memoriesforlife.org/

The Senior Companion System

The prototype developed for the Companions project (Wilks 2007, 2008; Wilks et al., 2008) contains three major functionalities, photographs, news and links to social networking sites and information sites (like Wikipedia[3]).

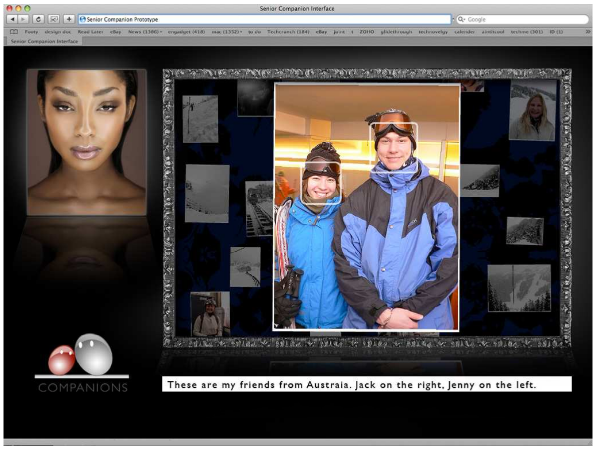
The main scope of the system is to elicit information about photographs and their content; where and when they were taken, details about the people in them and their relationship to the user and each other. The system keeps a record of the user's input and is capable of mapping it to a structured knowledge base. The system allows the use to perform some simple photo management including selecting photos by simply pointing and grouping different pictures together through dialogue.

Each photo loaded is processed with OpenCV[4], a computer vision library capable of identifying faces. This library provides to the SC face coordinates of all the people in the picture. This information is then used by the Dialogue Manager (DM) to refer to the position of the people appearing in the photograph. So, if there are three people in the picture next to each other, the system can ask questions about the person on the left, the one on the right or the one in the middle. The relevant positioning is obtained from the coordinates returned by the OpenCV system. Throughout the conversation, the system will ask about both spatial and temporal attributes of the picture such as the occasion and place where the photo was taken. It will also delve into the possible relationships that exist between the people in the photo. Since all the sessions are stored, if a person appears in multiple photos and the system manages to recognise the user through the face recognition software, the SC is smart enough to realise that it already knows the person so no further questions are asked about that person. If a place is mentioned in a photo, e.g. Pisa, the system goes immediately to the Wikipedia site and derives a question about Pisa (such as "Did you visit the Leaning Tower?") to show knowledge of what it is doing.

The SC is also capable of switching to other functions such as reading the news. This feature is interesting because it showcases the ability of the system to handle more than one kind of application at a time. The news is obtained via RSS feeds supplied from the BBC news website. These news items can span any category be it politics, sports, business etc thus imposing another requirement on the SC, i.e. the ability of having an unconstrained vocabulary. While the SC is reading the news, the mixed initiative approach allows the user to start or stop the feed by simply speaking with the Companion.

Since a lot of information about the user is already on line and does not need to be elicited through conversation, the Companion can go to a social networking site such as Facebook and build an initial database of the User's friends and relatives. This information mainly consists of photographs with annotations through which the social network of the person can be identified. This allows the SC to learn information about the user without asking for information about a person when this information is explicitly defined in Facebook.

The modular approach of the basic system allows the SC to expand further and in so doing, generate more conversational data for machine learning research. The system architecture and modules are described briefly below.


Senior Companion Interface
Senior Companion Prototype
Google
Footy design doc Read Later eBay News (1386) engadget (418) mac (1352) to do Techcrunch (184) eBay joint t ZOHO glidethrough technovelgy calender aintitcool techme (301) ID (1)
Senior Companion Interface
COMPANIONS

These are my friends from Australia. Jack on the right, Jenny on the left.

Figure 1 : Senior Companion Interface

The Senior Companion is made up of:

- A visually appealing interface with a realistic avatar which acts as a mediator between the system and the user.
- Multi-modal input such as speech, pointing, keyboard and touch.
- Face identification and recognition software capable of identifying face locations and similarities.
- Social Networking links which accepts pre-annotated (XML) photos as a means for creating richer dialogues quickly.
- A conversational module which talks with the user about topics within the photo domain: when and where the photo was taken, discussion of the people in the photo including their relationships to the user. It can also span beyond using the hybrid-world model by querying search engines online.
- A news module which can read the news from the following categories: politics, business and sports.
- Jokes telling feature which makes use of internet-based jokes website.
- A fully integrated Knowledge base for maintaining user information which contains:
    - A mechanism for storing information in a triple store (Subject-Predicate-Object)-the RDF Semantic Web format for handling

[3] http://www.wikipedia.org/
[4] http://sourceforge.net/projects/opencv/

    unexpected user input that falls outside of the photo domain, e.g. arbitrary locations in which photos might have been taken.

    - Ontological information which is exploited by the Dialogue Manager and provides domain-specific relations between fundamental concepts.

- A reasoning module which help the system infer new facts from the Knowledge Base and from the world knowledge obtained in RDF from the Internet.

- Contains basic photo management capability allowing the user in conversation to select photos as well as display a set of photos with a particular feature.

System Architecture

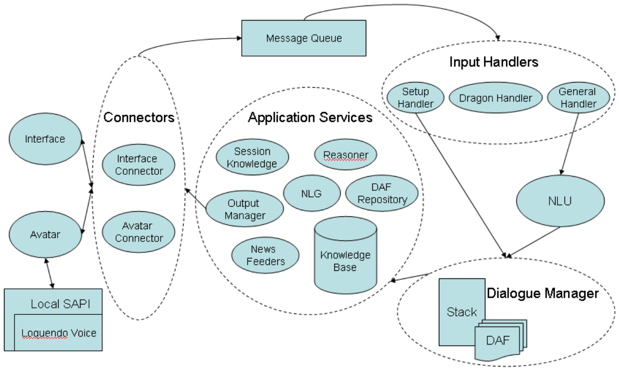


Figure 2 : Senior Companion system architecture

The SC architecture can be seen in Figure 2 and is divided into three abstract level components – Connectors, Input Handlers and Application Services together with the Dialogue Manager.

Connectors act as a bridge between the main system and the external applications (such as the avatars, interface, etc). These external applications are similar to plug-ins which can be easily added, removed or exchanged at will. A single connector exists between the system and each external application and this acts as a simple interface which abstracts the underlying complex communication protocol details. This approach makes the management of further modules much easier. At the moment, the system has two connectors implemented; the Napier Interface Connector (from COMPANIONS consortium partners at Napier University) and the CrazyTalk Avatar[5] Connector.

Input Handlers are modules that take care of the messages sent from the handlers and pass them on to other modules for processing. Each handler deals with a category of messages where categories are coarse-grained and could include one or more message types. The handlers act as an abstraction which separate the code handling inputs into different places and make the code easier to locate and change. In the current implementation of the SC, three handlers have been implemented; Setup Handler, Dragon Events Handler and General Handler. The Setup Handler takes care of the initial loading of the photo annotations (if they exist), it also performs the face detection and checks with the Knowledge Base if the photo being processed has been discussed in earlier sessions. Dragon Event Handler deals with Dragon speech recognition commands sent from interface while the General Handler processes user utterances and photo change events of the interface. The Dragon Naturally Speaking system has performed well, and a single user scenario is perfectly appropriate to a Companion design.

Application Services is a set of internal modules which provide an interface for the Dialogue Action Forms (DAF). The DAFs are in fact ATNs (Augmented Transition Networks) originally designed by Woods (Woods 1980) for syntactic analysis, structures that embody both hierarchical sub-networks (i.e. RTNs in automata theory) as well as augmentations on the arcs between nodes which enable any operations to be performed, within the dialogue process or offline in the knowledge base or multimodal world. The creation of these DAFs is obtained through an intuitive graphical editor designed to help the DAF designers manage the flow and associate tests and actions to the DAF. It acts as the communication link between DAFs and the internal system and enables DAFs to access system functionalities. The other modules in the Application Services include the News Feeder, the DAF Repository, the Natural Language Generation (NLG), the Session Knowledge, the Knowledgebase, the Reasoner and the Output Manager.

The News Feeder access a set of RSS feeders and gathers news items. The current implementation makes use of the BBC website to gather information about Sports, Politics and Business. Apart from this, there is also an extension which allows the system to occasionally relate a joke downloaded from a Jokes Feeder which works in a similar way. The DAF repository is a library of DAFs loaded from files generated by using the DAF editor mentioned earlier. Given a name, the repository simply returns a DAF which is stored on disk. The NLG module is responsible for formatting the replies using natural language. It randomly selects a system utterance from a template which is then presented to the user as the reply of the system. The module also accepts variables as input, these variables are used to mould the answer returned by the system. For example, if we are talking about a person in one of the photographs called John, the NLG (after being directed by the Dialogue Manager) will get a template for the next question such as “How is X related to you?” and it will use the parameter (which in this case is John) to replace X. Thus, the final answer of the NLG module will be “How is John related to you?”. The Session Knowledge is the place where global information for a particular running session is stored. For example, the name of the user who is running the session, the list of photos being discussed in this session, the list of user utterances, etc. The Knowledge Base is an RDF triplestore based upon the

[5] http://www.reallusion.com/crazytalk/

Jena framework[6] which is used to store the persistent knowledge. The triplestore API is actually a layer built upon a traditional relational database in our case we are using a MySQL[7] database. The application can save/retrieve information as RDF triples rather than table records. The structure of knowledge represented in RDF triples is discussed later. The Reasoner is used to perform inference on existing knowledge in the Knowledge Base. Finally, the Output Manager deals with all the messages sent to the external applications mentioned earlier. It has been implemented in a publisher/subscriber fashion. In the current implementation, there are three different channels in the system – the text channel, the interface command channel and the avatar command channel. Those channels could be subscribed by any connectors and handled respectively.

The Hybrid-World approach

The original work behind the SC was based on a closed world where the user converses with the system. The system instigates further conversation and in so doing elicits the discovery of tacit knowledge from the user. Whilst conducting the initial tests, the limitations of this approach immediately became evident. As soon as the user spoke with the SC, the conversation quickly went through unexpected paths which required more knowledge than was stored within the knowledge-base. For example, if an elderly person is speaking with the SC about an old photograph taken during WWII, the person would easily recall events of the period. Initially, the only approach we had was to make use of a chatbot since the bot could easily keep the conversation generic without understanding the specific events. However, conversing with a chatbot is not an inspiring experience so we modified our approach to make use of information stored on the web. The problem with that is that the web is an open world and it is quite hard for an agent to locate information, assimilate it in its knowledge base and use it.

Our Hybrid-World approach tackles these issues. Initially, it makes use of a closed-world where all the information is stored in the knowledge-base. Every utterance is passed through the Natural Language Understanding (NLU) module for processing. This module uses a set of well-established Natural Language processing tools such as the GATE (Cunningham, et al., 1997) system. The basic processes carried out by GATE are: tokenizing, sentence splitting, POS tagging, parsing and Named Entity Recognition. These components have been further enhanced for the SC system by adding new and improved gazetteers. These include amongst others new locations and family relationships. The NE recognizer is a key part of the NLU module and recognizes the significant entities required to process dialogue in the photo domain: PERSON NAMES, LOCATION NAMES, FAMILY_RELATIONS and DATES. Although GATE recognises basic entities, more complex entities are not handled. Because of this, apart from the gazetteers mentioned earlier and the hundreds of extraction rules already present in GATE, about 20 new extraction rules using the JAPE rule language were also developed for the SC module. These included rules which identify complex dates, family relationships, negations and other information related to the SC domain. The following is an example of a simple rule used to identify relationship in utterances such as “Mary is my sister”:

```
Macro: RELATIONSHIP_IDENTIFIER
(
({Token.category=="PRP$"}|{Token.category=="PRP"}|{L
ookup.majorType=="person_first"}):person2
({Token.string=="is"})
({Token.string=="my"}):person1
({Lookup.minorType=="Relationship"}):relationship
)
```

Using this rule with the example mentioned earlier, the rule interprets person1 as referring to the speaker thus if the name of the speaker is John (which was known from previous conversations) it is utilised. Person 2 is the name of the person. This name is recognised by using the gazetteers we have in the system (which contain about 40,000 first names). Apart from this, it can also refer to a preposition which can be easily disambiguated using the anaphora resolver found in GATE. The relationship is once again identified using the almost 800 unique relationships added to the gazetteer.

With this information, the NLU module identifies Information Extraction patterns in the dialogue that represent significant content with respect to a user's life and photos. The NLU module also identifies a Dialogue Act Tag for each user utterance based on the DAMSL set of DA tags and prior work done jointly with the University of Albany (Webb et al., 2008).

The information obtained (such as Mary sister-of John) is passed to the Dialogue Manager (DM) and then stored in the knowledge base (KB). The DM filters what to include and exclude from the KB. If, for example, the NLU module discovered that Mary is the sister of John, the NLU knows that sister is a relationship between two people and as such, it is a very important piece of information. However, the NLU also discovers a lot of syntactical information such as the fact the both Mary and John are nouns. Even though this information is important, it is too low level to be of any use by the SC with respect to the user, i.e. the user is not interested in the parts-of-speech of a word. Thus, this information is discarded by the DM and not stored in the KB.

Once the information is filtered by the DM, the KB stores the information. In essence, the KB is a long term store of information which makes it possible for the SC to retrieve information stored between different sessions. The information can be accessed anytime it is needed by simply invoking the relevant calls. The structure of the data in the database is an RDF triple. This is why it is more commonly refer to as a triple store. In mathematical terms, a triple store is nothing more than a large database of interconnected graphs. Each triple is made up of a subject, a predicate and an object. So if we had to take the previous example, Mary sister-of John; *Mary* would be the subject, *sister-of* would be the predicate and *John* would be the object. If we had to imagine this graphically, Mary and John would be two distinct points in a 3D space and the sister-of relationship would be the line (or relationship) that joins these two points

[6] http://jena.sourceforge.net/

[7] http://www.mysql.com/

in space. There are various advantage to using this structure; first, the relationship between different objects is explicitly defined using the predicates in the triples. The second advantage is that it is very easy to perform inferences on such data. So if in our knowledgebase, we add a new triple which states that Tom is the son of Mary, we can easily infer (by using the previous facts) that John is the uncle of Tom.

Uncle Inference Rule:

```
        (?a sisterOf ?b),
        (?x sonOf ?a),
        (?b gender male) -> (?b uncleOf ?x)
```

Triples:

```
        (Mary   sisterOf        John)
        (Tom    sonOf           Mary)
```

Triples produced automatically by ANNIE (the semantic tagger):

```
        (John   gender          male)
```

Inference:

```
        (Mary   sisterOf        John)
        (Tom    sonOf           Mary)
        (John   gender          male)
        ->
         (John uncleOf Tom)
```

This kind of inference is already used by the SC and in fact, we do have about 50 inference rules aimed at producing new data on the relationships domain. This combination of triple store, inference engine and inference rules makes a powerful system which mimics human reasoning and thus gives the SC a sense of intelligence. For our prototype we are using the JENA Semantic Web Framework for the inference engine together with a MySQL database as the knowledgebase.

Even though this approach is very powerful, it is still not enough to cover all the possible topics which can crop up during a conversation. So in such circumstances, rather than switching over to a chat-bot, the DM switches to an open-world model and instructs the NLU to seek further information online.

The NLU makes use of different approaches to achieve this. When the DM requests further information on a particular topic, the NLU first checks with the KB whether the topic is about something known. At this stage, we have to keep in mind that any topic requested by the DM should be already in the KB since it was preprocessed by the NLU when it was mentioned in the utterance. As an example, if the user informs the system that the photograph was taken in "Paris", the utterance is first processed by the NLU and then sent to the DM. Once the DM requests further information about "Paris", the NLU goes through the KB and retrieves any triples related to "Paris". Typically, ANNIE (A Nearly New Information Extraction engine), our semantic tagger, would have already identified "Paris" as a location and this information would be stored in the KB. If it is not found, the semantic tagger analysis the topic and provides the NLU with the missing information.

Once the type of the information is identified, the NLU would use the various strategies predefined inside it. In the case of locations, one of these strategies would be to seek for information in Wiki-Travel[8] or Virtual Tourists[9]. The system already knows how to query these sites and interpret their output by using predefined wrappers. A wrapper is essentially a file which describes where a particular piece of information is located. This is then used to extract that information from the webpage. So a query is sent online to these sites and the information retrieved is stored in the triple-store. This information is then used by the DM to generate a reply. In the previous example, the system manages to extract the best sightseeing spots in Paris. The NLU would then store in the KB triples such as [Paris, sight-seeing, Eiffel Tower] and the DM with the help of the NLG would ask the user "I've heard that the X is a very famous spot. Have you ever seen it while you were there?" Obviously in this case, X will be replaced by the "Eiffel Tower".

On the other hand, if the topic requested by the DM is unknown or the semantic tagger is not capable of understanding the semantic category, the system makes use a normal search engine. A query is sent to these search engines and the top pages are retrieved. These pages are then processed using ANNIE and the different attributes are analysed. The standard attributes returned by ANNIEinclude information about Dialogue Acts, Polarity (i.e. whether a sentence has positive, negative or neutral connotations), Named Entities, Semantic Categories (such as dates and currency), etc. The system then filters the information collected by using generic patterns and generates a reply from the resultant information. So if the user is talking about cats, the system searches for cats online. It processes the pages and its current strategy is to identify all the statements by using Dialogue Acts. So in our example, the system would retrieve the following statements:

- Cats may be the most popular pet in the world
- Cats recover quickly from falls
- Some people don't like Persian Cats

These statements are then checked for polarity and only the most prevailing statements are kept (i.e. if the statements are prevailingly negative then the system will give a negative answer, so on and so forth). In this example, the first two statements are prevailingly positive because of words such as "popular" and "recover" so the answer returned will be a positive one. The NLU would then select one of these two statements at random, send it to the DM and using the NLG, it would reply "You know that I've heard that X" where X is replaced with "cats may be the most popular pet in the world".

In synthesis, this hybrid world approach allows us to focus on the closed world that exists between the user and the system but when necessary, the system is allowed to venture cautiously in the open world thus enriching the user experience.

[8] http://wikitravel.org/

[9] http://www.virtualtourist.com/

Conclusion and Future Work

The SC prototype utilizes a hybrid-world approach which enriches the interaction between the user and the system. The features mentioned above have been implemented and tested. This approach proved to be superior to the closed world approach because it does not limit the system to the information inside its own databases derived directly from the user in conversation; the general aim here is to break out of the classic AI paradigm of reasoning with strong systems over limited closed worlds. We are aiming at a system with weak representation and reasoning (i.e. RDF). On the other hand, it is a manageable approach and does not suffer for the problems which an open world model would have. Our approach is a lazy approach whereby a closed-world model is used but if necessary; our agents are allowed to venture beyond our systems onto the open world and harvest information which might be useful for the progress of the conversation.

What we have so far is simply an initial platform on which to build something more interesting and complex during the latter half of the project with our partners. During the first two years a number of prototypes have been developed and evaluated and the SC is just one of those, though so far the best performing. The Consortium will now build an integrated demonstrator with at least the SC functionality we have described; hopefully adding in during the third year full face recognition and using machine learning to give a much more flexible barrier between the closed (comfortable) and open (uncomfortable) worlds. Moreover, we are working on a strong emotional and cognitive models (based around Ekman, 1999, Wundt, 1913, Cowie, Douglas-Cowie, Savvidou, & McMahon, 2000) so that the state of the Companion can be expressed both in language and other modalities. When dealing with understanding problems, this companion will be able to interface with both closed or open worlds thus exploiting the best of both worlds.

Acknowledgement

This work was entirely funded by the Companions project (www.companions-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

References

Cowie, R., Douglas-Cowie, E., Savvidou, S., McMahon, E., Sawey, M. and Schroder, S. (2000). 'FEELTRACE': An instrument for recording perceived emotion in real time. ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion, Newcastle, UK .

Cunningham et al. (1997). Cunningham, H., Humphreys, K., Gaizauskas, R., Wilks, Y., GATE -- a TIPSTER based General Architecture for Text Engineering. In Proceedings of the TIPSTER Text Program (Phase III) 6 Month Workshop. DARPA, Morgan Kaufmann, California

Ekman, P. (1999). Basic Emotions. Handbook of Cognition and Emotion, Dalgleish, T. and Power, M. (Eds.) John Wiley, New York.

Lauria, S. (2007). Talking to Machines: Introducing Robot Perception to Resolve Speech Recognition Uncertainties. Circuits Systems Signal Processing vol 26(4) pp. 513-526.

Lee, Chul Min, Narayanan, Shrikanth S. / Pieraccini, Roberto (2002): "Combining acoustic and language information for emotion recognition", In *ICSLP-2002*, 873-876.

O’Reilly, T (2005). What is Web 2.0. In O’Reilly Media Conference, 2005

Webb, N., Liu, T., Hepple, M., and Wilks, Y. (2008). Cross Domain Dialogue Act Tagging. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-08). Marrakech, Morocco, 2008

Wilks, Y. (2007) Has there been progress on talking sensibly to computers ? Science, Volume: 318

Wilks, Y. (2008) The Semantic Web and the Apotheosis of annotation. In Proc. IEEE Intelligent Systems.(May/June 2008)

Wilks, Y., Catizone, R., and Mival, O. (2008), The Companions paradigm as a method for eliciting and organising life data, In Proc. Workshop on Memories for Life, British Computer Society, London, March 18, 2008

Woods, W. (1980) Cascaded ATN grammars. American Journal of Computational Linguistics, 6(1):1—12.

A Set of Collaborative Semantics for an Abstract Dialogue Framework *

Julieta Marcos and **Marcelo A. Falappa** and **Guillermo R. Simari**
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur
Email: {mjm, mfalappa, grs}@cs.uns.edu.ar

Abstract

We present an *Abstract Dialogue Framework* which provides an environment for studying the behavior of collaborative dialogue systems in terms of abstract relevance notions, together with three *Collaborative Semantics* each of which defines a different collaborative behavior of the dialogues under the framework. One of these semantics describes an utopian, non practical, behavior which is approximated in different ways by the other two constructive semantics. Complete examples are provided in Propositional Logic Programming.

1 Introduction and Motivation

This work is about modeling collaborative dialogues in multi-agent systems (MAS). By *collaborative* we mean that the agents are willing to share any relevant knowledge (to the topic at issue). The final objective of this research topic is the design of *formal dialogue systems* which follow an *adequate* behavior. To that end, a *formal specification* of such behavior (or possible acceptable behaviors) would be useful. Besides, there is a need of *practical behaviors* (suitable to be implemented in a distributed MAS consisting of autonomous agents which have access only to the private knowledge of their own and to the public knowledge generated in the dialogue) and, even better, whose specification is *constructive*.

Most of the existent works in the area propose a formal system for some particular type of dialogue, based upon certain reasoning model (mostly argumentative systems) and identify properties of the generated dialogues, usually termination and properties of the outcome, e. g. in [Kraus *et al.*, 1998], [Parsons *et al.*, 2002], [Amgoud *et al.*, 2005], [Amgoud *et al.*, 2007], [Black and Hunter, 2007]. We have observed that there are some desirable properties of these systems which are rarely satisfied. One such property is ensuring that, when the dialogue ends, there is no relevant information left unpublished, not even distributed among several participants. This property may not be easy to achieve if the underlying logic is complex. For example, argumentation-based dialogues usually consist of interchanging arguments *for* and *against* certain claim, but they do not consider other possible relevant contributions which are not necessarily arguments. In particular, in [Black and Hunter, 2007] this property is successfully achieved, but for a simplified version of a particular argumentative system (further details and comparison to that work are included in [Marcos *et al.*, 2009]). Another property which is in some cases overlooked is ensuring that the final conclusion is coherent with all what has been said during the dialogue.

These observations motivated the present work in which we intend to abstractly and formally specify the main requirements to be achieved by collaborative dialogue systems, as well as analyzing to what extent these can be fulfilled in a distributed environment where none of the participants has access to the entirety of the information. In this first approach, we will consider a restricted notion of collaborative dialogue which takes place among a fixed set of homogeneous agents equipped with finite and static knowledge bases expressed in a common knowledge representation language. The only possible move in the dialogue will be to make a contribution (to publish a subset of one's private knowledge base) and no other locution (such as questions, proposals, etc.) will be allowed. We will make no assumption regarding the nature of the underlying reasoning model, except for being a well defined function which computes a unique outcome, given a topic and a knowledge base. This will make our analysis suitable for a wide range of underlying logics, regardless whether they are monotonic or non-monotonic, and also including both argumentative and non-argumentative approaches.

2 Informal Requirements

We believe that an adequate behavior of collaborative dialogue systems should ideally satisfy the following:

R$_1$: All the **relevant** information is exposed in the dialogue.
R$_2$: The exchange of **irrelevant** information is avoided.
R$_3$: The final conclusion **follows** from all what has been said.

Thus, we will conduct our analysis of such behavior in terms of two abstract elements: a *reasoning model* and a *relevance notion* [1] assuming that the former gives a formal meaning to the word *follows*, and the latter to the word *relevant*. Both elements are domain-dependent and, as we shall see, they are not unattached concepts. It is important to mention that the

*This research is partially supported by Sec. Gral. de Ciencia y Tecnología (Univ. Nac. del Sur), CONICET and Agencia Nac. de Prom. Científica y Técnica (ANPCyT).

[1] Other works which make an explicit treatment of the notion of relevance in dialogue are for example [Parsons *et al.*, 2007] and [Prakken, 2001]. Further details and comparison to these works are included in [Marcos *et al.*, 2009].

relevance notion is assumed to work in a context of *complete information*. Also recall that our analysis will be intended to be suitable both for monotonic and non-monotonic logics.

We believe that the achievement of R_1-R_3 should lead to achieving other important requirements (listed below) and, hence, part of the contribution of this work will be to state the conditions under which this hypothesis actually holds.

$\mathbf{R}_4$: The dialogue should always end.

$\mathbf{R}_5$: Once the dialogue ends, if the agents added all their still private information, and reasoned from there, the previously drawn conclusions should not change.

In the task of simultaneously achieving requirements R_1 and R_2, in the context of a distributed MAS, a non-trivial problem arises: relevant information distributed in such a way that none of the parts is relevant by itself. A simple example illustrates this situation: suppose that A knows that a implies b, and also that c implies b, and B knows that a, as well as d, holds. If agents A and B engage in dialogue for determining whether b holds or not, then it is clear that the relevant information is: *a implies b*, and *a holds*. However, neither A knows that a holds, nor B knows that a implies b, making them unaware of the relevance of this pieces of information. It is true, though, that A could suspect the relevance of *a implies b* since the dialogue topic, b, is the consequent of the implication, but she has certainly no way of anticipating any difference between this and *c implies b*. This last means that, either she abstains from exposing any of the two implications (relegating R_1), or she tries with some or both of them (relegating R_2, in the case she chooses the wrong one first). In short, there is a tradeoff between requirements R_1 and R_2. Because of the nature of collaborative dialogues, we believe that R_1 may be mandatory in many application domains, and hence we will seek solutions which achieve it, even at the expense of relegating R_2 a bit. Although a concrete solution will depend on specific instances of the reasoning model and the relevance notion, we feel it is possible to analyze how could solutions be constructed for the abstract case. The basic idea will be to develop a new relevance notion (a *potential relevance notion*) able to detect parts of distributed relevant contributions (under the original notion). Furthermore, we will see how the concept of *abduction* in logic is related to the construction of this potential relevance notions.

The rest of this work is organized as follows: Section 3 introduces an *Abstract Dialogue Framework* useful for carrying out the abstract study of collaborative dialogues. In Section 4 we formalize requirements R_1-R_3 by defining an *Utopian Semantics* for the framework, and show why it is not in general implementable in a distributed MAS. In Section 5 we propose alternative, practical semantics which approximate the utopian behavior, by achieving one of the requirements, either R_1 or R_2, and relaxing the other. Examples throughout this work are given in Proposit. Logic Programming.

3 Abstract Dialogue Frameworks

Three languages are assumed to be involved in a dialogue: the *Knowledge Representation Language* $\mathcal{L}$ for expressing the information exchanged by the agents, the *Topic Language* $\mathcal{L}_T$ for expressing the topic that gives rise to the dialogue, and the *Outcome Language* $\mathcal{L}_O$ for expressing the final conclusion (or *outcome*). These languages will be kept abstract in our formal definitions, but for the purpose of examples they will be instantiated in the context of Propositional Logic Programming (*PLP*) and its extension with Negation As Failure (*PLP$_{naf}$*). It is also assumed a language $\mathcal{L}_I$ for agent identifiers. As mentioned in section 1 we consider a restricted notion of dialogue which is *based on contributions only*. The following is a *public view of dialogue*: agents private knowledge is not considered.

Definition 1 (Move). *A* move *is a pair $\langle id, \mathrm{X}\rangle$ where $id \in \mathcal{L}_I$ is the identifier of the* speaker*, and $\mathrm{X} \subseteq \mathcal{L}$ is her* contribution.

Definition 2 (Dialogue). *A* dialogue *is a tuple $\langle t, \langle m_j\rangle, o\rangle$ where $t \in \mathcal{L}_T$ is the* dialogue topic*, $\langle m_j\rangle$ is a sequence of moves, and $o \in \mathcal{L}_O$ is the* dialogue outcome.

As anticipated in Section 2, we will study the behavior of such dialogues in terms of two abstract concepts: *relevance* and *reasoning*. To that end, an *Abstract Dialogue Framework* is introduced, whose aim is to provide an environment under which dialogues take place. This framework includes: the languages involved in the dialogue, a set of participating agents, an abstract relevance notion and an abstract reasoning model. An *agent* is represented by a pair consisting of an agent identifier and a private knowledge base, providing in this way a *complete view* of dialogues.

Definition 3 (Agent). *An* agent *is a pair $\langle id, \mathrm{K}\rangle$, noted K_{id}, where $\mathrm{K} \subseteq \mathcal{L}$ is a private finite knowledge base, and $id \in \mathcal{L}_I$ is an agent identifier.*

A *relevance notion*, in this article, is a criterion for determining, given certain already known information and a topic, whether it would be relevant to add certain other information (*i.e.,* to make a contribution). We emphasize that this criterion works under an assumption of *complete information*, to be contrasted with the situation of a dialogue where each agent is unaware of the private knowledge of the others. This issue will be revisited in Section 4. A *reasoning model* will be understood as a mechanism for obtaining a conclusion about a topic, on the basis of an individual knowledge base.

Definition 4 (Abstract Dialogue Framework). *An* abstract dialogue framework *$\mathfrak{F}$ is a tuple $\langle \mathcal{L}, \mathcal{L}_T, \mathcal{L}_O, \mathcal{L}_I, \mathcal{R}_t, \Phi, \mathrm{Ag}\rangle$ where $\mathcal{L}$, $\mathcal{L}_T$, $\mathcal{L}_O$ and $\mathcal{L}_I$ are the languages involved in the dialogue, Ag is a finite set of agents, $\mathcal{R} \subseteq 2^{\mathcal{L}} \times 2^{\mathcal{L}} \times \mathcal{L}_T$ is an* abstract relevance notion*, and $\Phi : 2^{\mathcal{L}} \times \mathcal{L}_T \Rightarrow \mathcal{L}_O$ is an* abstract reasoning model*. The brief notation $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag}\rangle$ will be also used.*

Notation. *If $(\mathrm{X}, \mathrm{S}, t) \in \mathcal{R}$, we say that X is a* t-relevant contribution to S under $\mathcal{R}$*, and we note it $\mathrm{X}\mathcal{R}_t\mathrm{S}$. When it is clear what relevance notion is being used, we just say that X is a t-relevant contribution to S. For individual sentences α in $\mathcal{L}$, we also use the simpler notation $\alpha\mathcal{R}_t\mathrm{S}$ meaning that $\{\alpha\}\mathcal{R}_t\mathrm{S}$.*

Throughout this work we will make reference to the following partially instantiated dialogue frameworks. It is assumed that the reader is familiarized with the concept of derivation in *PLP* (noted $\vdash$) and *PLP$_{naf}$* (noted $\vdash_{naf}$).

- $\mathfrak{F}^{lp} = \langle \mathcal{L}^{lp}, \mathcal{L}_{Facts}, \{\mathrm{Yes}, \mathrm{No}\}, \mathcal{L}_I, \mathcal{R}_\mathrm{t}, \Phi^{lp}, \mathrm{Ag}\rangle$ where $\mathcal{L}^{lp}$ is the set of rules and facts in *PLP*, $\mathcal{L}_{Facts} \subset \mathcal{L}^{lp}$ is the subset of facts (which in this case works as the Topic Language) and $\Phi^{lp}(\mathrm{S}, h) = \mathrm{Yes}$ if $\mathrm{S} \vdash h$, and No otherwise.

- $\mathfrak{F}^{naf} = \langle \mathcal{L}^{naf}, \mathcal{L}_{Facts}, \{\text{Yes}, \text{No}\}, \mathcal{L}_I, \mathcal{R}_t, \Phi^{naf}, \text{Ag}\rangle$ where $\mathcal{L}^{naf}$ is the set of rules and facts in *PLP*$_{naf}$ and $\Phi^{naf}(\mathrm{S}, h) = \text{Yes}$ if $\mathrm{S} \vdash_{naf} h$, and No otherwise.

It is useful to notice the existence of two different sets of knowledge involved in a dialogue: the *private knowledge* which is the union of the agents' knowledge bases, and the *public knowledge* which is the union of all the contributions already made, up to certain step. The former is a static set, whereas the latter grows as the dialogue progresses.

Definition 5 (Public Knowledge)**.** *Let d be a dialogue consisting of a sequence* $\langle \langle id_1, \mathrm{X}_1\rangle \ldots \langle id_m, \mathrm{X}_m\rangle \rangle$ *of moves. The* public knowledge *associated to d at step j* ($j \leq m$) *is the union of the first j contributions of the sequence and is noted* $\mathbf{PU}^j_d$ ($\mathbf{PU}^j_d = \mathrm{X}_1 \cup \cdots \cup \mathrm{X}_j$)*.*

Definition 6 (Private Knowledge)**.** *Let* $\mathfrak{F}$ *be an abstract dialogue framework including a set* Ag *of agents. The* private knowledge *associated to* $\mathfrak{F}$ *(and to any admissible dialogue under* $\mathfrak{F}$*) is the union of the knowledge bases of the agents in* Ag*, and is noted* $\mathbf{PR}_{\mathfrak{F}}$ ($\mathbf{PR}_{\mathfrak{F}} = \bigcup_{\mathrm{K}_{id} \in \text{Ag}} \mathrm{K}$)*.*

In our restricted notion of dialogue, agents' contributions must be subsets of their private knowledge. We define a set of *admissible dialogues* under a given framework, as follows.

Definition 7 (Admissible Dialogues)**.** *Let* $\mathfrak{F} = \langle \mathcal{L}, \mathcal{L}_T, \mathcal{L}_O, \mathcal{L}_I, \mathcal{R}_t, \Phi, \text{Ag}\rangle$ *be an abstract dialogue framework,* $t \in \mathcal{L}_T$ *and* $o \in \mathcal{L}_O$*. A dialogue* $\langle t, \langle m_j\rangle, o\rangle$ *is* admissible *under* $\mathfrak{F}$ *if, and only if, for each move* $m = \langle id, \mathrm{X}\rangle$ *in the sequence, there is an agent* $\mathrm{K}_{id} \in \text{Ag}$ *such that* $\mathrm{X} \subseteq \mathrm{K}$*. The set of admissible dialogues under* $\mathfrak{F}$ *is noted* $d(\mathfrak{F})$*.*

Remark. *Note that for any step j of any dialogue* $d \in d(\mathfrak{F})$*, it holds that* $\mathbf{PU}^j_d \subseteq \mathbf{PR}_{\mathfrak{F}}$*.*

Returning to the notions of relevance and reasoning, it was mentioned in Section 2 that these were not unattached concepts. Intuitively, a coherent dialogue must exhibit some connection between them. A natural connection is to consider that a contribution is relevant if its addition alters the conclusion achieved by the reasoning model, as defined bellow.

Definition 8 (Natural Relevance Notion)**.** *Let* Φ *be an abstract reasoning model. The* natural relevance notion *associated to* Φ *is a relevance notion* $\mathcal{N}^{\Phi}_t$ *defined as follows:* $\mathrm{X}\mathcal{N}^{\Phi}_t\mathrm{S}$ *iff* $\Phi(\mathrm{S}, t) \neq \Phi(\mathrm{S} \cup \mathrm{X}, t)$*. If* $\mathrm{X}\mathcal{N}^{\Phi}_t\mathrm{S}$*, we say that* X *is a* natural t-relevant contribution to S under Φ.

It will be seen later that this connection can be relaxed, *i.e.,* other relevance notions which are not *exactly* the natural one, will also be accepted. We distinguish the subclass of abstract dialogues frameworks in which the relevance notion is the natural one associated to the reasoning model. We refer to them as *Inquiry Dialogue Frameworks* [2] , and the relevance notion is omitted in their formal specification.

Definition 9 (Inquiry Dialogue Framework)**.** *An abstract dialogue framework* $\mathfrak{I} = \langle \mathcal{R}_t, \Phi, \text{Ag}\rangle$ *is an* Inquiry Dialogue Framework *if, and only if, it holds that* $\mathcal{R}_t = \mathcal{N}^{\Phi}_t$*. The brief notation* $\mathfrak{I} = \langle \Phi, \text{Ag}\rangle$ *will be used.*

Throughout this work we will make reference to the natural relevance notions $\mathcal{N}^{lp}_h$ and $\mathcal{N}^{naf}_h$, associated to the reasoning models Φ^{lp} and Φ^{naf}, and also to the inquiry frameworks $\mathfrak{I}^{lp}$ and $\mathfrak{I}^{naf}$, which result from $\mathfrak{F}^{lp}$ and $\mathfrak{F}^{naf}$, by instantiating the abstract relevance notions with the natural ones.

4 Utopian Collaborative Semantics for Abstract Dialogue Frameworks

A *semantics* for an abstract dialogue framework, in this work, is a subset of the admissible dialogues, whose elements satisfy certain properties, representing a particular *dialogue behavior*. In Section 2 we identified three requirements, R_1-R_3, to be ideally achieved by collaborative dialogue systems. In this section we will define an *Utopian Collaborative Semantics* which gives a formal characterization of such ideal behavior in terms of the elements of the framework.

In order to translate requirements R_1-R_3 into a formal specification, some issues need to be considered first. In particular, the notion of *relevant contribution* needs to be adjusted. On the one hand, there may be contributions which does not qualify as relevant but it would be adequate to allow. To understand this, it should be noticed that, since relevance notions are related to reasoning models, and reasoning models may be non-monotonic, then it is possible for a contribution to contain a relevant subset, without being relevant itself. Consider, for instance, the following set of rules and facts in the context of the $\mathfrak{I}^{naf}$ framework: $\{\ a \leftarrow b \wedge \mathbf{not}\ c,\quad b\ \}$, which is a natural a-relevant contribution to the empty set, but if we added the fact c, then it would not. In these cases, we say that the relevance notion fails to satisfy *left-monotonicity* and that the whole contribution is *weakly relevant* [3].

Definition 10 (Left Monotonicity)**.** *Let* $\mathcal{R}_t$ *be a relevance notion. We say that* $\mathcal{R}_t$ *satisfies* left monotonicity *iff the following condition holds: if* $\mathrm{X}\mathcal{R}_t\mathrm{S}$ *and* $\mathrm{X} \subseteq \mathrm{Y}$ *then* $\mathrm{Y}\mathcal{R}_t\mathrm{S}$*.*

Definition 11 (Weak Contribution)**.** *Let* $\mathcal{R}_t$ *be a relevance notion. We say that* X *is a* weak t-relevant contribution *to* S *iff the following holds: there exists* $\mathrm{Y} \subseteq \mathrm{X}$ *such that* $\mathrm{Y}\mathcal{R}_t\mathrm{S}$*.*

On the other hand, there may be contributions which qualify as relevant but they are not *purely* relevant. Consider, for example, the following set in the context of any of the two instantiated inquiry frameworks: $\{\ a \leftarrow b,\quad b,\quad e\ \}$, which is a natural a-relevant contribution to the empty set, although the fact e is clearly irrelevant. These impure relevant contributions must be avoided in order to obey requirement R_2. For that purpose, *pure relevant contributions* impose a restriction over weak relevant ones, disallowing absolutely irrelevant sentences within them, as defined bellow.

Definition 12 (Pure Contribution)**.** *Let* $\mathcal{R}_t$ *be a relevance notion, and* X *a weak t-relevant contribution to* S*. We say that* X *is a* pure t-relevant contribution *to* S *iff the following condition holds for all* $\alpha \in \mathrm{X}$*: there exists* $\mathrm{Y} \subset \mathrm{X}$ *such that* $\alpha\mathcal{R}_t(\mathrm{S} \cup \mathrm{Y})$*.*

Finally, it has been mentioned that the relevance notion works under an assumption of *complete information*, and thus it will be necessary to inspect the private knowledge of the others for determining the actual relevance of a given move.

[2] The term *Inquiry* is inspired on the popularized typology of dialogues proposed in [Walton and Krabbe, 1995].

[3] The term *weak relevance* is used in [Prakken, 2005] in a different sense, which should not be related to the one introduced here.

Now we are able to give a formal interpretation of requirements R_1-R_3 in terms of the abstract framework elements:

Definition 13 (Utopian Collaborative Semantics)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. A dialogue $d = \big\langle t, \langle m_j \rangle, o \big\rangle \in d(\mathfrak{F})$ belongs to the* Utopian Collaborative Semantics *for $\mathfrak{F}$ (noted Utopian($\mathfrak{F}$)) iff:*

Correctness: *if m_j is the last move in the sequence, then $\Phi(\mathbf{PU}^j_d, t) = o$.*

Global Progress: *for each move $m_j = \langle id_j, \mathrm{X}_j \rangle$ in the sequence, there exists $\mathrm{Y} \subseteq \mathbf{PR}_{\mathfrak{F}}$ such that $\mathrm{X}_j \subseteq \mathrm{Y}$ and Y is a pure t-relevant contribution to $\mathbf{PU}^{j-1}_d$.*

Global Completeness: *if m_j is the last move in the sequence, then $\mathbf{PR}_{\mathfrak{F}}$ is not a weak t-relevant contribution to $\mathbf{PU}^j_d$.*

Requirement R_3 is achieved by the *Correctness* condition, which states that the dialogue outcome coincides with the application of the reasoning model to the public knowledge at the final step of the dialogue. Requirement R_2 is achieved by the *Global Progress* condition, which states that each move in the sequence is part of a distributed pure relevant contribution to the public knowledge generated so far. Finally, requirement R_1 is achieved by the *Global Completeness* condition, which states that there are no more relevant contributions, not even distributed among different knowledge bases, after the dialogue ends. An illustrative example is given next.

Example 1. *Consider an instance of the $\mathfrak{I}^{lp} = \langle \Phi^{lp}, \mathrm{Ag} \rangle$ framework, with the set* Ag *composed by: $\mathrm{K_A} = \{a \leftarrow b, e\}$, $\mathrm{K_B} = \{b \leftarrow c, b \leftarrow d, f\}$, and $\mathrm{K_C} = \{c, g\}$. The following dialogue d_1, over topic a, and also all the permutations of its moves with the same topic and outcome, belong to the Utopian Semantics for the framework. The chart bellow traces the dialogue, showing the partial results of reasoning from the public knowledge so far generated. The last of these results (underlined) is the final dialogue outcome:*

$d_1 =$

step	A	B	C	$\Phi(\mathbf{PU}^{\mathrm{step}}_{d_1}, a)$
1	$a \leftarrow b$			No
2		$b \leftarrow c$		No
3			c	<u>Yes</u>

An essential requirement of dialogue systems is ensuring the termination of the generated dialogues. This is intuitively related to requirement R_2 (achieved by *global progress*) since it is expected that agents will eventually run out of relevant contributions, given that their private knowledge bases are finite. This is actually true as long as the relevance notion satisfies an intuitive property which states that a relevant contribution must add some new information to the public knowledge.

Definition 14 (Novelty)**.** *A relevance notion $\mathcal{R}_t$ satisfies* novelty *iff the following condition holds: if $\mathrm{X}\mathcal{R}_t\mathrm{S}$ then $\mathrm{X} \nsubseteq \mathrm{S}$.*

Then it can be ensured that any dialogue satisfying *global progress* under a relevance notion which satisfies novelty, *terminates* after a finite sequence of steps.

Proposition 1 (Termination)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework, and $d = \big\langle t, \langle m_j \rangle, o \big\rangle \in d(\mathfrak{F})$. If the notion $\mathcal{R}_t$ satisfies* novelty *and dialogue d satisfies* global progress *under $\mathfrak{F}$, then $\langle m_j \rangle$ is a finite sequence of moves.*

Another desirable property of these collaborative systems is ensuring it is not possible to draw different conclusions, for the same set of agents and topic. In other words, from the entirety of the information, it should be possible to determine the outcome of the dialogue, no matter what sequence of steps are actually performed. Furthermore, this outcome should coincide with the result of applying the reasoning model to the private knowledge involved in the dialogue. We emphasize that this is required for *collaborative* dialogues (and probably not for non-collaborative ones). For instance, in Example 1 the conclusion achieved by all the possible dialogues under the semantics is Yes which is also the result of reasoning from $\mathrm{K_A} \cup \mathrm{K_B} \cup \mathrm{K_C}$. This is intuitively related to requirements R_1 (achieved by *global completeness*) and R_3 (achieved by *correctness*) since it is expected that the absence of relevant contributions implies that the current conclusion cannot be changed by adding more information. This is actually true as long as the relevance notion is the natural one associated to the reasoning model, or a *weaker* one, as stated bellow.

Definition 15 (Stronger Relevance Notion)**.** *Let $\mathcal{R}_t$ and $\mathcal{R}'_t$ be two relevance notions. We say that the notion $\mathcal{R}_t$ is* stronger *than the notion $\mathcal{R}'_t$ iff the following holds: if $\mathrm{X}\mathcal{R}_t\mathrm{S}$ then $\mathrm{X}\mathcal{R}'_t\mathrm{S}$ (i.e., $\mathcal{R}_t \subseteq \mathcal{R}'_t$). We will also say that $\mathcal{R}'_t$ is* weaker [4] *than $\mathcal{R}_t$.*

Proposition 2 (Outcome Determinism)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework and $d = \big\langle t, \langle m_j \rangle, o \big\rangle \in d(\mathfrak{F})$. If d satisfies* correctness *and* global completeness *under $\mathfrak{F}$, and $\mathcal{R}_t$* is weaker than *$\mathcal{N}^{\Phi}_{t}$, then $o = \Phi(\mathbf{PR}_{\mathfrak{F}}, t)$.*

For example, in *PLP*, a relevance notion which detects the generation of new derivations for a given literal, would be weaker that the natural one. It is easy to see that this weaker relevance notion would also achieve *outcome determinism*.

The following corollaries summarize the results for the Utopian Semantics. For the case of inquiry frameworks, it is easy to see that any natural relevance notion satisfies *novelty*.

Corollary 1. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. The dialogues in Utopian($\mathfrak{F}$) satisfy* termination *and* outcome determinism*, provided that the relevance notion $\mathcal{R}_t$ satisfies* novelty *and is* weaker *than $\mathcal{N}^{\Phi}_t$.*

Corollary 2. *Let $\mathfrak{I}$ be an inquiry framework. The dialogues in Utopian($\mathfrak{I}$) satisfy* termination *and* outcome determinism*.*

It is clear that Definition 13 of the Utopian Semantics is not constructive, since both *global progress* and *global completeness* are expressed in terms of the private knowledge $\mathbf{PR}_{\mathfrak{F}}$, which is not entirely available to any of the participants. Example 2 shows that it is not only not constructive, but also in many cases not even implementable in a distributed MAS.

Example 2. *Consider the $\mathfrak{I}^{lp}$ framework instantiated in Example 1, and the following dialogue d_2:*

$d_2 =$

step	A	B	C	$\Phi(\mathbf{PU}^{\mathrm{step}}_{d_2}, a)$
1	$a \leftarrow b$			No
2		$b \leftarrow d$		No
3		$b \leftarrow c$		No
4			c	<u>Yes</u>

Dialogue d_2 does not belong to the Utopian Semantics because step 2 violates the global progress *condition. However,*

[4] Observe that here we use the term *weaker*, as the opposite of *stronger*, denoting a binary relation between relevance notions, and this should not be confused with its previous use in Definition 11 of *weak relevant contribution*.

it would not be possible to design a dialogue system which allows d_1 (presented in Example 1) but disallows d_2, since agent B *can not know in advance that c, rather than d, holds.*

The undesired situation is caused by a relevant contribution distributed among several agents in such a way that none of the parts is relevant by itself, leading to a tradeoff between requirements R_1 and R_2 (*i.e.,* between *global progress* and *global completeness*). In the worst case, each sentence of the contribution resides in a different agent. Thus, to avoid such situations, it would be necessary for the relevance notion to warrant that every relevant contribution contains at least one individually relevant sentence. When this happens, we say that the relevance notion satisfies *granularity*, defined bellow.

Definition 16 (Granularity)**.** *Let $\mathcal{R}_t$ be a relevance notion. We say that $\mathcal{R}_t$ satisfies* granularity *iff the following condition holds: if $\mathrm{X}\mathcal{R}_t\mathrm{S}$ then there exists $\alpha \in \mathrm{X}$ such that $\alpha\mathcal{R}_t\mathrm{S}$.*

Unfortunately, the relevance notions we are interested in, fail to satisfy granularity. It does not hold in general for the natural notions associated to deductive inference mechanisms. It has been shown in Example 2 that it does not hold for the simple case of *PLP*, and clearly neither PLP_{naf}.

5 Practical Collaborative Semantics for Abstract Dialogue Frameworks

The lack of *granularity* of relevance notions motivates the definition of alternative semantics which approach the utopian one, and whose distributed implementation is viable. The simplest approach is to relax requirement R_1 by allowing distributed relevant contributions to be missed, as follows.

Definition 17 (Basic Collaborative Semantics)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. A dialogue $d = \langle t, \langle m_j \rangle, o \rangle \in d(\mathfrak{F})$ belongs to the* Basic Collaborative Semantics *for $\mathfrak{F}$ (noted Basic$(\mathfrak{F})$) iff the following conditions, as well as* **Correctness***, hold:*

Local Progress: *for each move $m_j = \langle id_j, \mathrm{X}_j \rangle$ in the sequence, X_j is a pure t-relevant contribution to $\mathbf{PU}_d^{j-1}$.*

Local Completeness: *if m_j is the last move in the sequence, then it does not exist an agent $\mathrm{K}_{id} \in \mathrm{Ag}$ such that* K *is a weak t-relevant contribution to $\mathbf{PU}_d^j$.*

Requirement R_2 is achieved by *local progress* which states that each move in the sequence constitutes a pure relevant contribution to the public knowledge generated so far. Notice that this condition implies *global progress* (enunciated in Section 4). Requirement R_1 is now compromised. The *local completeness* condition states that each agent has no more relevant contributions to make after the dialogue ends. It is easy to see that, unless the relevance notion satisfies *granularity*, this is not enough for ensuring *global completeness* (enunciated in Section 4). As a result, requirement R_4 (termination) is achieved, given the same condition as in Section 4, whereas requirement R_5 (outcome determinism) cannot be warranted. These results are summarized in the corollary bellow.

Corollary 3. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. The dialogues in Basic$(\mathfrak{F})$ satisfy* termination *provided that the relevance notion $\mathcal{R}_t$ satisfies* novelty.

Considering the same scenario as in Example 1 for the $\mathfrak{I}^{lp}$ framework, it is easy to see that the only possible dialogue under the Basic Collaborative Semantics is the empty one (*i.e.,* no moves are performed), with outcome = No. More interesting examples, of dialogues with more than one step under the Basic Semantics, can be developed in the context of nonmonotonic reasoning models (see [Marcos *et al.*, 2009]).

In Section 2 we argued that requirement R_1 may be mandatory in many domains, but the Basic Semantics does not achieve it unless the relevance notion satisfies *granularity*, which does not usually happen. In order to make up for this lack of *granularity*, we propose to build a new notion (say $\mathcal{P}$) based on the original one (say $\mathcal{R}$) which ensures that: in the presence of a distributed relevant contribution under $\mathcal{R}$, *at least one* of the parts will be relevant under $\mathcal{P}$. We will say that $\mathcal{P}$ is a *potential relevance notion for* $\mathcal{R}$, since its aim is to detect contributions that could be relevant within certain *context*, but it is uncertain whether that context actually exists or not. Observe that the *context* is given by other agents' private knowledge which has not been exposed yet. Bellow we define the binary relation (*"is a potential for"*) between relevance notions, and also its propagation to dialogue frameworks.

Definition 18 (Potential Relevance Notion)**.** *Let $\mathcal{R}_t$ and $\mathcal{P}_t$ be relevance notions. We say that $\mathcal{P}_t$ is a* potential (relevance notion) *for $\mathcal{R}_t$ iff the following conditions hold: (1) $\mathcal{R}_t$ is stronger than $\mathcal{P}_t$, and (2) if $\mathrm{X}\mathcal{R}_t\mathrm{S}$ then there exists $\alpha \in \mathrm{X}$ such that $\alpha\mathcal{P}_t\mathrm{S}$. If $\mathrm{X}\mathcal{P}_t\mathrm{S}$ and $\mathcal{P}_t$ is a potential for $\mathcal{R}_t$, we say that* X *is a* potential t-relevant contribution *to* S *under $\mathcal{R}_t$.*

Definition 19 (Potential Dialogue Framework)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ and $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, \mathrm{Ag} \rangle$ be abstract dialogue frameworks. We say that $\mathfrak{F}^*$ is a* potential (framework) *for $\mathfrak{F}$ iff $\mathcal{P}_t$ is a potential relevance notion for $\mathcal{R}_t$.*

Returning to the semantics definition, the idea is to use the potential framework under the Basic Semantics, resulting in a new semantics for the original framework. The following definition introduces the *Full Collaborative Semantics* which is actually a family of semantics: each possible potential framework defines a different semantics of the family.

Definition 20 (Full Collaborative Semantics)**.** *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework. A dialogue $d = \langle t, \langle m_j \rangle, o \rangle \in d(\mathfrak{F})$ belongs to the* Full Collaborative Semantics *for $\mathfrak{F}$ (noted Full$(\mathfrak{F})$) iff $d \in$ Basic$(\mathfrak{F}^*)$ for some framework $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, \mathrm{Ag} \rangle$ which is a potential for $\mathfrak{F}$. We will also use the more specific notation $d \in$ Full$(\mathfrak{F}, \mathcal{P}_t)$.*

In this way, each agent would be able to autonomously determine that she has no more *potential relevant contributions* to make, ensuring there cannot be any distributed relevant contribution when the dialogue ends, and hence achieving R_1. In other words, achieving *local completeness* under the potential relevance notion implies achieving *global completeness* under the original one, as stated bellow.

Proposition 3. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ and $\mathfrak{F}^* = \langle \mathcal{P}_t, \Phi, \mathrm{Ag} \rangle$ be abstract dialogue frameworks such that $\mathfrak{F}^*$ is a potential for $\mathfrak{F}$, and $d \in d(\mathfrak{F})$. If dialogue d satisfies* local completeness *under $\mathfrak{F}^*$ then it satisfies* global completeness *under $\mathfrak{F}$.*

Requirement R_2 is in now compromised, since the context we have mentioned may not exist. In other words, achieving *local progress* under the potential relevance notion does not ensure achieving *global progress* under the original one. The challenge is to design *good* potential relevance notions which considerably reduce the amount of cases in which a contribution is considered potentially relevant but, eventually, it is not.

Observe that a relevance notion which considers any sentence of the language as relevant, works as a potential for any given relevance notion, but it is clearly not a good one. Next we summarize the results for the dialogues generated under the Full Collaborative Semantics. By achieving *global completeness*, these dialogues achieve *outcome determinism* under the same condition as before. Although *global progress* is not achieved under the original relevance notion, it is achieved under the potential one, and thus *termination* can be ensured as long as the latter satisfies *novelty*.

Corollary 4. *Let $\mathfrak{F} = \langle \mathcal{R}_t, \Phi, \mathrm{Ag} \rangle$ be an abstract dialogue framework and $\mathcal{P}_t$ a potential for $\mathcal{R}_t$. The dialogues in Full($\mathfrak{F}, \mathcal{P}_t$) satisfy* termination *and* outcome determinism*, provided that $\mathcal{P}_t$ satisfies* novelty *and $\mathcal{R}_t$ is* weaker *than $\mathcal{N}_t^{\Phi}$.*

In order to provide a concrete example for the Full Collaborative Semantics, in the context of the $\mathfrak{I}^{lp}$ framework, we must first define a potential relevance notion for the corresponding natural one. The basic idea is to detect contributions that would be relevant given a certain *context of facts* (which are currently uncertain). To that end, we define the *abduction set* [5] associated to a given fact h and a given set S. In short, the abduction set of h from S is the set of all the minimal sets of facts that could be added to S in order to derive h.

Definition 21 (Abduction Set)**.** *Let $\mathrm{S} \subseteq \mathcal{L}^{lp}$ and $h \in \mathcal{L}_{Facts}$. The* abduction set *of h from S is defined as follows:*

$$AB(\mathrm{S}, h) = \{\mathrm{H} \subseteq \mathcal{L}_{Facts} : (\mathrm{S} \cup \mathrm{H}) \vdash h \textit{ and } \nexists \mathrm{H}' \subset \mathrm{H} \textit{ s. t. } (\mathrm{S} \cup \mathrm{H}') \vdash h\}$$

Next we introduce an *abductive relevance notion* $\mathcal{A}_h^{lp}$. Basically, X is an h-relevant contribution to S under this notion iff its addition generates a new element in the abduction set of h. This means that either a new fact-composed natural h-relevant contribution to S arises, or h is actually derived.

Definition 22 (Abductive Relevance)**.** *Let $\mathrm{S} \subseteq \mathcal{L}^{lp}$ and $h \in \mathcal{L}_{Facts}$. A set $\mathrm{X} \subseteq \mathcal{L}^{lp}$ is an h-relevant contribution to S under $\mathcal{A}_h^{lp}$ iff there exists $\mathrm{H} \subseteq \mathcal{L}_{Facts}$ such that: (1) $\mathrm{H} \in AB(\mathrm{S} \cup \mathrm{X}, h)$ and (2) $\mathrm{H} \notin AB(\mathrm{S}, h)$.*

It can be shown that $\mathcal{A}_h^{lp}$ is a potential for $\mathcal{N}_h^{lp}$ (see [Marcos *et al.*, 2009]). Illustrative examples of the dialogues generated under the Full Collaborative Semantics are given next.

Example 3. *Consider the same scenario as in Example 1 for the $\mathfrak{I}^{lp}$ framework. Both dialogues d_1 and d_2, presented in Example 1 and Example 2 respectively, belong to Full($\mathfrak{I}^{lp}, \mathcal{A}_h^{lp}$). Also belongs to this semantics the dialogue d_3 traced bellow. The fifth column of the chart shows the evolution of the abduction set of the fact a from the generated public knowledge. An additional step* ***0*** *is added, in order to show the initial state of this abduction set (i.e., when the public knowledge is still empty). Dialogue d_3 results from dialogue d_2 by interchanging steps* ***2*** *and* ***3****:*

$d_3 =$

step	A	B	C	$AB(\mathbf{PU}_{d_3}^{\mathrm{step}}, a)$	$\Phi(\mathbf{PU}_{d_3}^{\mathrm{step}}, a)$
0				$\{\{a\}\}$	No
1	$a \leftarrow b$			$\{\{a\}\{b\}\}$	No
2		$b \leftarrow c$		$\{\{a\}\{b\}\{c\}\}$	No
3		$b \leftarrow d$		$\{\{a\}\{b\}\{c\}\{d\}\}$	No
4			c	$\{\{\}\}$	$\underline{\text{Yes}}$

Also belongs to Full($\mathfrak{I}^{lp}, \mathcal{A}_h^{lp}$) the dialogue which results from d_2 by merging steps ***2*** *and* ***3*** *together in a single one. Note that all these dialogues achieve* global completeness*, although* global progress *is achieved only by dialogue d_1.*

6 Conclusions and Future Work

We proposed a possible characterization of collaborative dialogue systems' ideal behavior, in terms of two abstract elements: a *reasoning model* and a *relevance notion*. Then we showed the main problem which disallows the implementation of this utopian behavior as a distributed system: the presence of *distributed relevant contributions* in such a way that none of the parts is relevant by itself. We identified the cause of the problematic situation, which is the lack of *granularity* of relevance notions, and performed a further analysis reducing the problem to the task of designing an appropriate *potential relevance notion*. We showed a complete example in Propositional Logic Programming which makes use of abduction for designing such notion. In addition, we stated the conditions under which termination and accuracy of conclusions (outcome determinism) can be ensured.

As future work, we plan to: (1) study potential relevance notions for the case of more complex logic formalisms; (2) extend the present analysis to non-collaborative dialogue types (such as persuasion and negotiation); and (3) explicitly address the inconsistency problem that may arise in a dialogue when merging knowledge of different agents.

References

[Amgoud *et al.*, 2005] L. Amgoud, H. Prade, and S. Belabbes. Towards a formal framework for the search of a consensus between autonomous agents. *AAMAS'2005, Utrecht*, 2005.

[Amgoud *et al.*, 2007] L. Amgoud, Y. Dimopoulos, and P. Moraitis. A unified and general framework for argumentation based negotiation. *AAMAS'2007, Honolulu, Hawai'i*, 2007.

[Black and Hunter, 2007] E. Black and A. Hunter. A generative inquiry dialogue system. *AAMAS'2007, Honolulu, Hawai'i*, 2007.

[Kakas *et al.*, 1992] A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *J. Log. Comput.*, 1992.

[Kraus *et al.*, 1998] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 1998.

[Marcos *et al.*, 2009] J. Marcos, M.A. Falappa, and G.R. Simari. Abstract semantics for collaborative inter-agent dialogues. Technical report, LIDIA, UNS, Bahía Blanca, 2009.

[Parsons *et al.*, 2002] S. Parsons, L. Amgoud, and M. Wooldridge. An analysis of formal inter-agent dialogues. *AAMAS'2002, Bologna, Italy*, 2002.

[Parsons *et al.*, 2007] S. Parsons, P. McBurney, E. Sklar, and M. Wooldridge. On the relevance of utterances in formal inter-agent dialogues. *AAMAS'2007, Honolulu, Hawai'i*, 2007.

[Prakken, 2001] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 2001.

[Prakken, 2005] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *J. Log. Comput.*, 2005.

[Walton and Krabbe, 1995] D. Walton and E.C.W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.

[5] *Abduction* has been widely used for finding *explanations* for a certain result. A survey on the extension of Logic Programming to perform abductive reasoning is provided in [Kakas *et al.*, 1992].

Automatic handling of Frequently Asked Questions using Latent Semantic Analysis

Patrik Larsson and **Arne Jönsson**
Department of Computer and Information Science
Linköping University, SE-581 83, LINKÖPING, SWEDEN
patla073@gmail.com, arnjo@ida.liu.se

Abstract

We present results from using Latent Semantic Analysis (LSA) for automatic handling of FAQs (Frequently Asked Questions). FAQs have a high language variability and include a mixture of technical and non-technical terms. LSA has a potential to be useful for automatic handling of FAQ as it reduces the linguistic variability and capture semantically related concept. It is also easy to adapt for FAQ. LSA does not require any sophisticated linguistic analyses and merely involves various vector operations. We evaluate LSA for FAQ on a corpus comprising 4905 FAQ items from a collection of 65000 mail conversations. Our results show that Latent Semantic Analysis, without linguistic analyses, gives results that are on par other methods for automatic FAQ.

Introduction

Automatic FAQ-systems allow clients' requests for guidance, help or contact information to be handled without human intervention, c.f. (Åberg 2002). Typically, automatic FAQ-systems use previously recorded FAQ-items and various techniques to identify the FAQ-item(s) that best resembles the current question and present a matching answer. For instance, the FAQFinder system (Mlynarczyk and Lytinen 2005) uses existing FAQ knowledge bases to retrieve answers to natural language questions. FAQFinder utilises a mixture of semantic and statistical methods for determining question similarities. Another technique is to use a frequency-based analysis from an ordinary FAQ list with given/static questions and answers (Ng'Ambi 2002). Linguistic based automatic FAQ systems often starts with finding the question word, keywords, keyword heuristics, named entity recognition, and so forth (Moldovan et al. 1999). Another approach is to use machine learning techniques, such as support vector machines to predict an appropriate response (Marom and Zukerman 2007; Bickel and Scheffer 2004). Marom and Zukerman also utilise a variety of clustering techniques to produce more accurate answers.

One issue for automatic help-desk systems is that we often have many-to-many mappings between requests and responses. A question is stated in many ways and, as humans answer the requests, the response to a question can be stated in many ways. The propositional content can also vary, although operators re-use sentences, at least in e-mail help desk-systems (Zukerman and Marom 2006).

Help-desk e-mail conversations are further characterised by: (1) having many requests raising multiple issues, (2) having high language variability and (3) with many answers utilising non-technical terms not matching technical terms in the requests (Marom and Zukerman 2007).

In this paper we present results from experiments on using linear algebra techniques for automatic FAQ for single issues. We will not consider (1), i.e. we will not present answers to requests comprising multiple issues. Our study is based on a log of email dialogues between customers and help-desk operators at Hewlett-Packard (Marom and Zukerman 2007)[1]. A typical example is seen in Figure 1. The dialogues deal with a variety of issues such as technical assistance, Figure 1, product information, and product replacement.

Latent Semantic Analysis

To handle the FAQ problem of having high language variability and a mixture of non-technical terms and technical terms, it is natural to investigate techniques that in various ways reduce the linguistic variability and better capture semantically related concepts. One prominent such technique is vector space models, for an overview see Eldén (2007). The basic idea is to formulate the problem in a matrix terminology, usually by constructing a matrix using text units as columns and letting the words in all text units each have a row in the matrix. Then a certain entry in the matrix is nonzero iff the word corresponding to the row exists in the text unit represented by the column. The resulting matrix is very large and sparse. Techniques for reducing dimensionality are therefore often used.

Latent Semantic Analysis (LSA) (Landauer and Dumais 1997) is one of the most well-known dimensionality reduction vector space techniques and has also been used on a limited help-desk data set with promising results (Caron 2000).

LSA uses Singular Value Decomposition (SVD) to find a reduced vector space that fits the original as well as possible using a lower ranked matrix. The SVD factorizes a matrix A

[1] We are indebted to HP, Yuval Marom and Ingrid Zukerman for providing this corpus.

User question:

```
When I start my computer from time to time I get the following error message
""Terminating thread due to stacking overflow problem.  A VxD, possibly recently installed,
has consumed too much space.
Increase setting of MinSPs' in System.INI or remove recent VxD's.
4 SP's installed""
Please tell me how to make the change in number SP's installed to correct this problem.
Thanks, FirstName
```

Model answer:

```
RE: Armada 1500 P-prtbl-dg" "Dear HP Customer,
Thank you for contacting HP eServices.
This is ni response to your e-mail regarding the error message on your Armada 1500 Notebook.
Please be informed that Stack overflow conditions are indicative of errors in the device
driver.  Device drivers are allocated 4 kilobytes (one page) of space to be used as a
stack.  If the device driver uses more than 4 kilobytes of memory, a stack overflow condition
occurs.
In order to resolve the issue we recommend following the steps given below.
```

A long list of instructions follows, which are removed.

```
We recommend visiting the following link for more information on the error message.
```

Link removed

```
Please e-mail us, if you need any further assistance and we will be glad to help.
Thank you,
HP eServices
```

Figure 1: Example of question and answer from the HP corpus.

into $A = USV^T$ where V^T is the new orthonormal basis for A, S is a diagonal matrix denoting how prevalent each column is in the basis while U is the coordinates of the original documents using the new basis.

Documents can then be compared in the new vector space, by changing the basis of the document to V^T and then compare likeness with the coordinates in U, often based on the cosine between the two.

LSA resolves problems with synonymy, polysemy, homonymy etc. by mapping (or mixing) terms occuring often in the same context to each other (Landauer et al. 2007).

For automatic FAQ systems LSA directly allows for mappings between combinations of questions and answers. The first-order relations in this domain are:

- Terms in questions – Terms in similar questions
- Terms in questions – Terms in questions with similar responses
- Terms in responses – Terms in similar responses
- Terms in responses – Terms in responses with similar questions

A ”request term” like *power coord* and similar terms used in other requests will be mapped to the technical term *AC-adapter* used in responses by the helpdesk-support personnel. ”Request terms” like *strange, blinking, green, light* will be mapped to the terms in other requests or responses resolving the issue at hand.

LSA also captures higher-order relations between terms, and thus create mappings between terms that do not directly co-occur, but that mutually co-occur with other terms.

LSA for automatic FAQ

When performing LSA on FAQs, a Question-Answer item (QA-item), such as Figure 1 in the corpus corresponds to a document. In the corpus questions and answers are simple text files with indicators for question and answer. The corpus comprise two-turn dialogues as well as longer dialogues with follow-up questions. Just as Marom and Zukerman 2007 we only used two-turn dialogues with reasonably concise answers (16 lines at most).

The vector space is constructed by having all QA-items in the matrix on one axis and all the words on the other and then calculate the frequency of the words in relation to the QA-items. Questions and answers are not separated in the QA-items. In our case we create the $m \times n$ matrix A where each matrix element a_{ij} is the weight of word i in document j. The n columns of A represent the QA-items in the corpus and the rows correspond to the words, as seen in Figure 2. We use 4414 QA-items for training comprising 35600 words. The size of the original training matrix A is thus 4414×35600. About 524900 elements of the total 157 millions are nonzero[2].

	QA$_1$	QA$_2$	...	QA$_n$
word$_1$	a_{11}	a_{12}	...	a_{1n}
word$_2$	a_{21}	a_{22}	...	a_{2n}
...	...	...	...	...
word$_m$	a_{m1}	a_{m2}	...	a_{mn}

Figure 2: A word–QA-item matrix

Performing SVD on A with dimension k produces three

[2] The exact numbers depend on which subset of QA-items that are used for training.

new matrices, the $m \times k$ matrix U which corresponds to the QA-items in the reduced vector space, the $k \times k$ diagonal matrix S and the $k \times n$ matrix V which corresponds to the words in the reduced vector space. The size of the matrix U depends on the dimension, k. For a reduced vector space with $k = 300$ it is 35600×300 and the size of the matrix V^T is 300×4414.

In order to do LSA for automatic FAQ we perform the following steps:

1. Pre-process all Question-Answer items in the corpus, Section ”Pre-processing”.
2. Perform Singular Value Decomposition on the matrix A_{tr} obtained from the set of QA-items in the training set. This gives us the three components U_{tr}, S_{tr} and V_{tr}^T.
3. Fold in the answers from the training set of QA-items into the set of vectors in U_{tr}, Section ”Folding Questions and Answers into LSA space”, Equation 4. This gives us a new matrix, U_{folded}, i.e. a *pseudo-document* with all answers folded into the reduced vector space.
4. Create answer clusters, $A_{cluster}$, from U_{folded} using QT-clustering, Section ”Answer clustering”.
5. Create a new matrix of tagged left singular vectors U_{tagged} by using the clusters $A_{cluster}$ to tag U_{tr} and remove items that do not belong to any cluster. Select a representative answer from each cluster, Section ”Selecting a representative answer from a faq-cluster”.
6. Fold in questions one by one from the test set, Section ”Folding Questions and Answers into LSA space”, Equation 5. Compare to the tagged matrix of left singular vectors U_{tagged}, see Section ”Answer clustering” and pick the best.

In what follows we will describe each step in more detail.

Pre-processing

The QA-items are used without any linguistic preprocessing, i.e. we use no stop-word lists, stemming, etc (Landauer et al. 2007). Nor any Named-Entity recognition or abbreviation lists.

The StandardAnalyzer in Lucene[3] is used for tokenization and vectorization, i.e. creating vectors from the tokens.

To reduce the impact of terms which are evenly distributed in the corpus, Question-Answer vectors are entropy normalised by using the global term weights from the matrix used for SVD, where a document consists of QA-items. These term weights are then used to weight the terms in the question and answer documents as follows (Gorrell 2006) :

$$p_{ij} = \frac{tf_{ij}}{gf_i} \tag{1}$$

$$gw_i = 1 + \sum_j \frac{p_{ij} log(p_{ij})}{log(n)} \tag{2}$$

$$c_{ij} = gw_i log(tf_{ij} + 1) \tag{3}$$

where c_{ij} is the cell at column i, row j in the corpus matrix and gw_i is the global weighting of the word at i, n is the number of QA-items. tf_j is the term frequency in document j and gf the global count of term i across all documents. Following Gorrell (2006) we use tf_{ij} instead of p_{ij} in Equation 3.

Performing Singular Value Decomposition

We use SVDLIBC[4] for singular value decomposition. Clustering and testing are performed in MatLab. SVDLIBC is used to generate the three components U_{tr}, S_{tr} and V_{tr}^T. We will investigate the influence of different dimensions, i.e. different reduced vector spaces.

We perform singular value decomposition on the training set of questions and answers. Answers are folded into the new, dimension reduced, vector space afterwards using Matlab, see Section ”Folding Questions and Answers into LSA space”.

The SVD components U_{tr}, S_{tr} and V_{tr}^T are imported to MatLab in Matlab ascii-format together with the files containing the training questions + training answers, the training answers, and later the test questions (and unique id-numbers for the dialogues).

Folding Questions and Answers into LSA space

Given that we have a vector space matrix A with QA-items and words but want to have answers as responses to requests we need to transform the answers into the reduced vector space. This is done by folding-in the answers into the reduced vector space model and produce a *pseudo-document* (Wang and Jin 2006).

The answers in the training corpus are folded into the reduced space after we performed SVD. This is in line with findings of Zukerman and Marom (Zukerman and Marom 2006) who find that using both questions and answers to retrieve an answer proved better than using only questions or answers.

Answers are folded in by taking the dot product of the vector with the reduced space right singular matrix, V_{tr}, i.e. the terms in the reduced vector space, Equation 4 (Gorrell 2006, p. 34).

$$\mathbf{a}_{folded} = \mathbf{a} \cdot V_{tr} \tag{4}$$

Taking all $\mathbf{a}_{folded}$ vectors creates U_{folded}, a *pseudo-document* representation of size $n(documents) \times k$ where k is the new reduced vector space dimension and $n(documents)$ is the number of terms in the QA-items, i.e. all unique words occuring in the questions and answers.

Similarly, questions in the test set need to be folded into the dimension reduced set of clustered answers, Equation 5, see Section ”Classifying new questions”.

$$\mathbf{q}_{folded} = \mathbf{q} \cdot V_{tr} \tag{5}$$

Folding in questions allows us to map questions to the reduced vector space.

[3] http://lucene.apache.org/

[4] http://tedlab.mit.edu/~dr/svdlibc/

```
Create a sparse matrix, $M$, with
  $m_{ij} = \begin{cases} 1 & \text{if } a_{ij} > \tau \\ 0 & \text{otherwise} \end{cases}$
  where
  $a_{ij} = a_i \cdot a_j$
  $a$ is an answer in the *pseudo-document* with the
  folded-in answers, $U_{folded}$, $i \neq j$
  and $\tau$ is the maximum cluster diameter
Extract clusters from $M$:
  **while** max(row-sum)> $\gamma$
    For each row $i$ in M calculate:
      $r_i = \sum_j a_{ij}$
    Save the row with highest row sum as cluster $c_k$
    Remove all rows and columns from $M$ belonging to
    cluster $c_k$
```

Figure 3: QT-clustering algorithm

Answer clustering

One problem with automatic handling of FAQ is that in the corpus one question can have many correct answers, depending on the person answering the question. Each operator uses different wordings. They also provide varying amounts of information (Zukerman and Marom 2006). Consequently, we want to cluster answers that provide the same, or similar, information. In the future, the use of FAQ-databases would alleviate this problem somewhat.

One way to handle this is to cluster around QA-items. However, the ”request domain” is a more open domain than the ”answer domain”, the ”request domain” often contains irony and completely irrelevant information, for example:

```
   If this gives you a good laugh that's
OK but I'm serious and very desperate -
at least I know that the CD isn't a cup
holder...  The number I have given you
is my home phone number.  There's no way
I'll take this call at work.
```

The answers on the other hand contain very specific instructions, are more formal in style and are devoid of irony, c.f. Figure 1. Thus, we only cluster the dialogues based on the answers.

We use Quality Threshold Clustering (Heyer, Kruglyak, and Yooseph 1999) for answer clustering, see Figure 3, and use cosine-distances between normalized answer vectors as the maximum cluster diameter, τ.

γ controls the number of elements in each cluster. A low γ may result in small clusters where the similarity of the answer is accidental, for example a user who by mistake submits the same question twice may receive two identical replies, the cluster consisting of these replies would not represent responses to a frequently asked question but merely the fact that the question was sent twice. A too low limit on cluster size therefore increases the risk of not including a relevant answer.

Creating the adjacency matrix, M, can be somewhat computationally demanding, but as it can be done incrementally it poses no computational problems.

This clustering method guarantees that the LSA-similarity of frequent answer clusters will not exceed a predefined threshold, and this threshold is meaningful because LSA-similarity between documents have shown a high correlation with human judgment (Landauer, Laham, and Foltz 1998).

A similarity threshold, τ, of 0.6 - 0.9 is usually considered acceptable, but it depends on the specific domain. We will investigate the best threshold, τ, for the FAQ domain. A technical domain like helpdesk-support might need a larger threshold than more ”soft domains”, as the answers are less varied. A large threshold generates large clusters which has an advantage in that there will be more questions and therefore more mappings between questions and answers. There will be more members in each cluster and also more clusters as there are more members (i.e. answers) that can be nearest neighbour when classified. Thus, we achieve a higher Coverage. On the other hand, a too large threshold probably means a decrease in Precision and Recall.

Classifying new questions

To find the best answer cluster for a new request we use a basic k-nearest neighbour classifier (Cardoso-Cachopo and Oliveira 2003). We perform the following steps:

1. Find the distance for the new request to the dimension reduced QA-items by computing the dot product of the new request, $\mathbf{q}$, with all U_{tr} vectors, $\mathbf{u}$, in all clusters, i.e. all QA-items.

   $a_i = \mathbf{q} \cdot \mathbf{u}_i$

2. Pick the k nearest a_i, i.e. answers close to the QA-items´, $\mathbf{u}$.

3. Select the cluster with most a_i items and a representative from that cluster answer as above, Section ”Selecting a representative answer from a faq-cluster”.

kNN is used to exclude outliers, QA-items that accidentally are close to the new request, e.g. QA-items containing misspelled words that are misspelled the same way in the new request.

Using a more sophisticated classifier might improve performance somewhat, but a basic classifier like kNN generally gives good performance when combined with Latent Semantic Analysis (Cardoso-Cachopo and Oliveira 2003).

Selecting a representative answer from a faq-cluster

To select an answer document from the matched cluster we first normalize the answer vectors to minimize the influence of ”flooded answers”, that is, answers that contain relevant information, but a large portion of irrelevant information as well (for example an answer message containing responses to more than one question). We use standard length normalisation:

$$\hat{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|} \tag{6}$$

A representative answer is then selected from, U_{tagged} using the cosine angel. Answers must exceed a threshold δ to be selected. This is done to ensure that they are not too far away from the request. We investigate two methods for selecting a representative answer. One method takes the answer closest to the centroid of the cluster as being the most representative answer. The other method takes the closest answer in the cluster. The former provides a more neutral answer and is probably not wrong but may not contain enough information. The latter, on the other hand, provides answers that may be wrong, but if correct probably convey more relevant information.

Evaluation

We have evaluated LSA for automatic FAQ on the corpus from HP with email messages between users and helpdesk operators. We use the same sub-corpus of 4,905 two-turn dialogues divided into 8 subsets as Marom and Zukerman (2007). In the experiments all 8 data sets are grouped into one large set. Typical for this test set is that answers are short (less than 16 lines). This was done to ensure that answers do not contain multiple answers etc. (Marom and Zukerman 2007). We use 4414 dialogues for training and 491 for testing.

We have conducted experiments to find optimal values of τ, SVD dimension and how to select an answer from the answer clusters; using the centroid in an answer cluster vs. taking the closest answer. We also study δ, k and the minimum cluster size γ.

We use the ROUGE tool set version 1.5.5 to produce Precision, Recall and F-scores for one-gram-overlaps (ROUGE-1). We apply equal weight to Recall and Precision when calculating F-scores. ROUGE then produces similar results as word-by-word measures (Marom and Zukerman 2007).

The term "Coverage" is used to measure the amount of a test set where any reply was given based on the threshold settings of the method used (Marom and Zukerman 2007).

Results and discussion

The output from the automatic FAQ system varies depending on from which data set an answer is retrieved. Some requests are answered using a short and fairly standardised answer which are easy to retrieve by the system. For instance finding answers to requests in the Product Replacement data set is mostly trivial, the documents are highly similar and can be matched on the basis of the title, Figure 5.

Other requests fail to produce an answer, or produce an empty response indicating that there are no answers close enough in the answer cluster. Many requests also produce correct, but not equal, answers as in Figure 6. In this case the answer contains more information than the original answer to the request did.

Parameter setting investigations

We have investigated the effect on different SVD dimensions, see Figure 7. As can be seen in Figure 7 the ROUGE-1 Precision, Recall and F-scores reach a maximum after a dimension of around 250 and stays the same up to around 650.

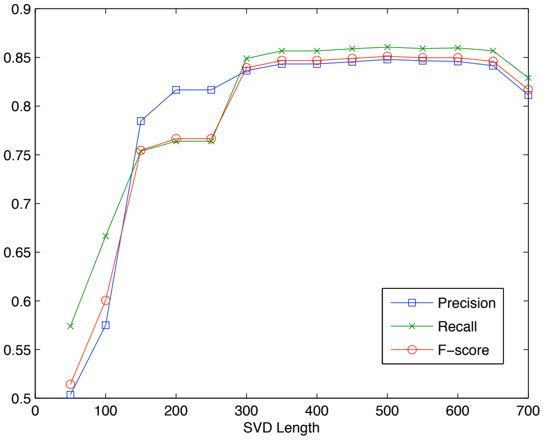


Figure 7: The influence of different SVD dimension reductions

We use the 300 first singular vectors as the gain from utilising a larger vector space does not motivate the longer processing time and increased memory usage needed for longer vectors. This is in line with previous findings by e.g. (Landauer and Dumais 1997).

The value on the threshold δ affects Coverage, normally we use $\delta = 0.6$ as this gives reasonable Coverage. When τ increases Coverage also increases. In the final investigation we use $\delta = 0.43$ when $\tau = 0.9$ to have the same Coverage of 29% as Marom and Zukerman (2007), see below.

We conducted experiments on the effect of using the centroid answer versus the answer closest to the request, max cosine. We did that for two different values of τ, as τ affect the number of answers in a cluster and consequently the centroid answer. Figure 4 shows values for Coverage, Precision, Recall and F-score for $\tau = 0.6$ and $\tau = 0.9$. Using the centroid answer gives Precision, Recall and F-scores that are higher than the corresponding values for closest answer for both values of τ. Coverage is slightly better, but that improvement does not justify the higher loss in Precision and Recall. We will, thus, use the centroid in the experiments presented below.

We have investigated the effect different values on k and γ have on Coverage, Precision and F-score, see Figure 9.

The parameters k in kNN and γ in QT-clustering co-varies. To study them one by one we used a fix value, 1, for k when varying γ and $\gamma = 1$ when varying k. To reduce the risk of equal votes, we only use odd values for k.

As can be seen in Figure 9 increasing γ and k have some effect up until $\gamma = 7$ and $k = 5$, for $k = 1$ and $\gamma = 1$ respectively. We use $k = 5$ and $\gamma = 5$ in our experiments. The parameters co-vary and the exact values are not critical, as long as they are not too small.

The QT-clustering diameter, τ, is varied between 0.6 and 0.9 (Landauer, Laham, and Foltz 1998). For small τ we get a higher Coverage, and slightly lower Precision, but Precision is not that much affected for τ between 0.6 and 0.9, Figure 8.

To be more precise. The ROUGE-1 values for τ = 0.7,

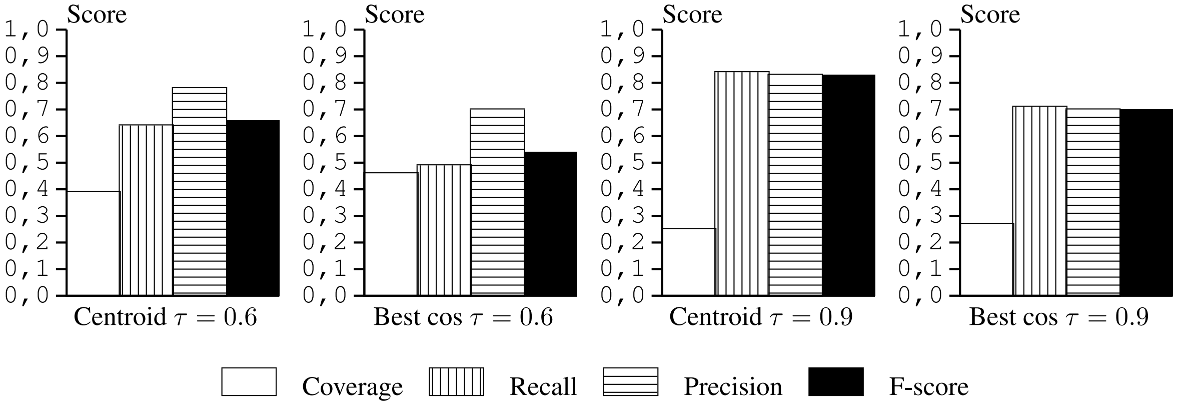


Figure 4: Answer selection based on the centroid vs the best cosine match.

Request, Q, to be answered	**Request corresponding to the answer A**
I need a replacement return shipping label P-eurp Need return shipping label and box for 30 gig laptop hard drive, p/n *Product number*	**I need a replacement return shipping label P-eurp** I had 4 separate power supplies sent to me and no longer have the boxes with the return labels. I need 4 return labels for the following case numbers including the one above. Thank you. *List of ID numbers*
Answer in FAQ base corresponding to Q	**Answer, A, generated to Q**
RE: I need a replacement return shipping label P-eurp Good Morning, I apologize for the delay in responding to your issue. Your request for a return airbill has been received and has been sent for processing. Your replacement airbill will be sent to you via email within 24 hours. Thank You, E Services	**RE: I need a replacement return shipping label P-eurp** Good Morning, I apologize for the delay in responding to your issue. Your request for a return airbill has been received and has been sent for processing. Your replacement airbill will be sent to you via email within 24 hours. Thank You, E Services

Figure 5: Example of a trivial response. The upper left side shows the new request to the automatic FAQ system, the lower left side shows the correct corresponding answer to that request. The lower right hand side shows the answer presented as a response to the new request, the request in the upper left, and the upper right shows the request that corresponds to the presented answer.

providing 35% Coverage, are:

Average_R: 0.71792	(95%-conf.int. 0.67195 - 0.76372)
Average_P: 0.79824	(95%-conf.int. 0.76347 - 0.83251)
Average_F: 0.72578	(95%-conf.int. 0.68454 - 0.76708)

Allowing a slightly higher Coverage of 40%, τ = 0.6, we still achieve acceptable results:

Average_R: 0.64948	(95%-conf.int. 0.60443 - 0.69585)
Average_P: 0.78956	(95%-conf.int. 0.75779 - 0.82156)
Average_F: 0.66915	(95%-conf.int. 0.62896 - 0.71028)

Comparisons to other approaches

Our results are better than the results obtained when using SVM only (Bickel and Scheffer 2004). For instance, for a Recall around 30% they never produce Precision above 0.7. However, they use a different corpus, so the results are not totally comparable.

It is not straightforward to compare our results with the results from Marom and Zukerman (2007). Marom and Zukerman present three different methods: *Doc-Pred, Sent-Pred* and *Sent-Hybrid*, and also a meta-learning method utilising combinations of the methods (Marom, Zukerman, and Japkowicz 2007). Basically *Doc-Pred* uses request features from the questions to retrieve the best complete answer (possibly comprising more than one sentence). *Sent-Pred* use the same technique to analyse the new request but here the answer is composed of sentences from different answers. *Sent-Hybrid*, finally, utilise terms in the requests and answers, to further improve performance.

We do not split answers into sentences, and consequently, we should not be able to reproduce the results from Marom and Zukerman (2007) where they produce answers com-

Request, Q, to be answered	Request corresponding to the answer A
iPAQ H3700 Series Pocket PC P-hand I am unable to download activesync 3.6. The Microsoft website is not working when I click the download button.	**iPAQ H3700 Series Pocket PC P-hand** I have a H3635 and the software that was on my computer to sync information from the computer to the cradle is now erased from my computer, how do I install new software on my computer? Can I install it from the internet site or do I have to purchase the CD from the store because I do not have the orginal one that came with my Pcket PC...Help me
Answer in FAQ base corresponding to Q	**Answer, A, generated to Q**
RE: iPAQ H3700 Series Pocket PC P-hand Dear Customer, Thank you for contacting HP eServices. Please visit the link given below and download Activesync 3.6: *web site* If you need any more clarifications, please email us back and we would be glad to help you. Regards, HP eServices	**RE: iPAQ H3700 Series Pocket PC P-hand** Dear HP Customer, Thank you for contacting HP eServices. If you need a synchronization software in order to synchronize the iPAQ with your Desktop PC, you could download the latest ActiveSync 3.6 from Microsoft"s website at *web site* For complete installation instructions of the above ActiveSync 3.6 software, please visit the link *web site* If you need any further clarifications, please e-mail us back and we would be glad to help. Thank you, HP eServices

Figure 6: Example of a correct but not identical response. The upper left side shows the new request to the automatic FAQ system, the lower left side shows the correct corresponding answer to that request. The lower right hand side shows the answer presented as a response to the new request, the request in the upper left, and the upper right shows the request that corresponds to the presented answer.

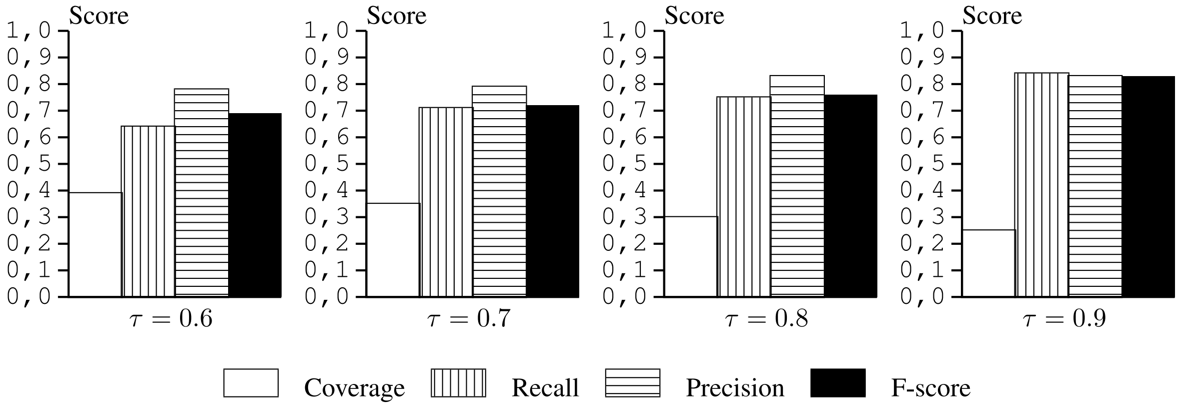


Figure 8: Comparison of different cluster diameters for $k = 5$ and $\delta = 0.6$.

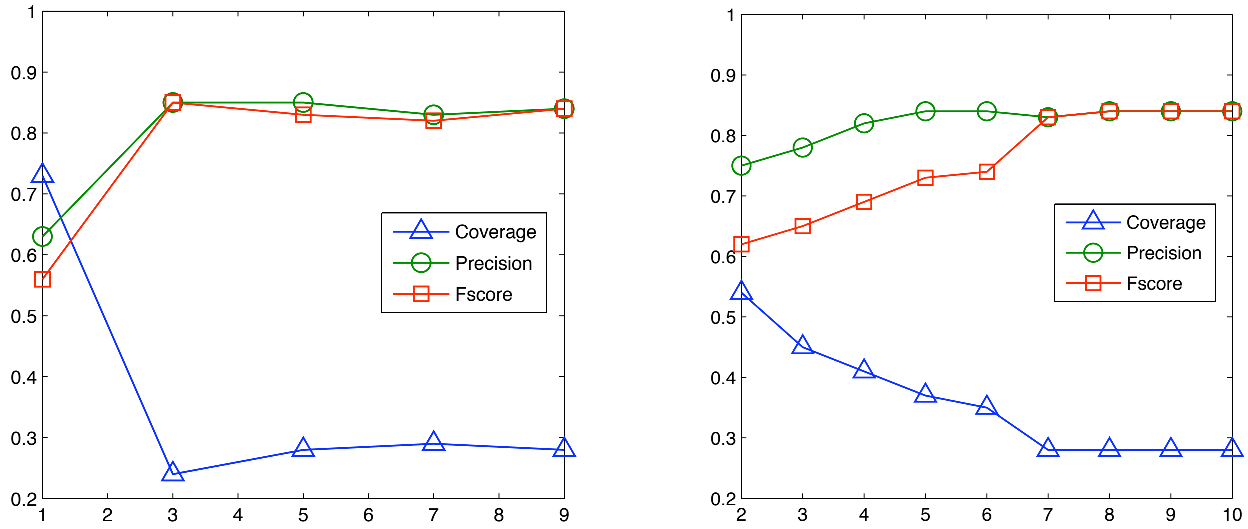
posed by sentences from different answers, *Sent-Pred, Sent-Hybrid*. Comparing our results with *Doc-Pred* we see that our results are similar to their results, Table 1[5].

To be more precise, we use τ = 0.9 and $\delta = 0.43$ to achieve 29% Coverage. With k = 5 we get the following ROUGE-1 values:

Average_R: 0.83814 (95%-conf.int. 0.80448 - 0.87163)
Average_P: 0.82713 (95%-conf.int. 0.79470 - 0.85939)
Average_F: 0.82726 (95%-conf.int. 0.79412 - 0.86162)

To further verify the results, we conducted a ten-fold evaluation on the whole corpus. This gave the following

[5] Values on *Doc-Pred* from Marom and Zukerman (2007).



k with $\gamma = 1$

γ with $k = 1$

Figure 9: The influence of different γ and k

	Coverage	Recall	F-score
Doc-Pred	29%	0.82	0.82
LSA	29%	0.83	0.83

Table 1: LSA compared to *Doc-Pred*

ROUGE-1 values (τ = 0.9, k = 5, 29% Coverage):

```
Average_R: 0.81231   (95%-conf.int. 0.79795 - 0.82571)
Average_P: 0.85251   (95%-conf.int. 0.84344 - 0.86194)
Average_F: 0.80643   (95%-conf.int. 0.79401 - 0.81870)
```

We see that the there is a small decrease in Recall and a slight increase in Precision. The reason for this is that there are a number of empty messages that give 100% Precision and 0% Recall. The results are, however, still on par with *Doc-Pred*.

Summary

In this paper we have presented results from using Latent Semantic Analysis for automatic FAQ handling. Using LSA is straightforward and requires very little domain knowledge or extra processing steps such as identifying terms, removing stop words, etc. All we do are standard vector operations, mainly in LSA space. Consequently, the method is easy to utilise in new domains.

Our results show that LSA is a promising method for automatic FAQ. The results are on a par with the *Doc-Pred* method of Marom and Zukerman (2007).

One problem with LSA is the computational demands of SVD. For practical applications it is possible to handle the computational problem with SVD by collecting Question-Answer pairs continuously and fold them into LSA space (clustering can be done incrementally), and update the SVD regularly (perhaps once a month) with ”representative” Question-Answer pairs used for mapping new questions to the domain.

Another possibility is to perform the SVD incrementally by using Generalised Hebbian Learning (GHA) for SVD (Gorrell 2006). This allows for incremental SVD and handles very large data sets. Yet another possibility is to reduce the dimensionality of the matrix on which SVD is calculated using Random Indexing (Gorrell 2006; Kanerva, Kristofersson, and Holst 2000; Sellberg and Jönsson 2008).

Further work includes splitting up answers into sentences and perform answer clustering like *Sent-Hybrid* (Marom and Zukerman 2007). By using sentences instead of answers in our matrix we can form answer clusters.

We did not perform any pre-processing as suggested by Landauer et al. (2007). Named-Entity recognition can probably further improve the results in a final system.

Acknowledgment

This research is financed by Santa Anna IT Research Institute AB.

References

Bickel, S., and Scheffer, T. 2004. Learning from message pairs for automatic email answering. In Boulicaut, J.-F.; Esposito, F.; Giannotti, F.; and Pedreschi, D., eds., *Proceedings of Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004*, volume 3201 of *Lecture Notes in Computer Science*, 87–98. Springer.

Cardoso-Cachopo, A., and Oliveira, A. L. 2003. An empirical comparison of text categorization methods. In *Inter-*

national Symposium on String Processing and Information Retrieval, SPIRE, LNCS, volume 10.

Caron, J. 2000. Applying lsa to online customer support: A trial study. Master's thesis, University of Colorado, Boulder.

Eldén, L. 2007. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial & Applied Mathematics (SIAM).

Gorrell, G. 2006. *Generalized Hebbian Algorithm for Dimensionality Reduction in Natural Language Processing*. Ph.D. Dissertation, Linköping University.

Heyer, L. J.; Kruglyak, S.; and Yooseph, S. 1999. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research* 9(11):1106–1115.

Kanerva, P.; Kristofersson, J.; and Holst, A. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society. Erlbaum, 2000.*, 1036.

Landauer, T. K., and Dumais, S. T. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104:211–240.

Landauer, T. K.; McNamara, D.; Dennis, S.; and W., K., eds. 2007. *Handbook of Latent Semantic Analysis*. Mahwah NJ: Lawrence Erlbaum Associates.

Landauer, T. K.; Laham, D.; and Foltz, P. 1998. Learning human-like knowledge by singular value decomposition: A progress report. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Marom, Y., and Zukerman, I. 2007. A predictive approach to help-desk response generation. In Veloso, M. M., ed., *Proceedings of IJCAI 2007, Hyderabad, India*, 1665–1670.

Marom, Y.; Zukerman, I.; and Japkowicz, N. 2007. A meta-learning approach for selecting between response automation strategies in a help-desk domain. In *AAAI*, 907–912. AAAI Press.

Mlynarczyk, S., and Lytinen, S. 2005. Faqfinder question answering improvements using question/answer matching. In *Proceedings of L&T-2005 - Human Language Technologies as a Challenge for Computer Science and Linguistics*.

Moldovan, D. I.; Harabagiu, S. M.; Pasca, M.; Mihalcea, R.; Goodrum, R.; Girju, R.; and Rus, V. 1999. Lasso: A tool for surfing the answer net. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8), 1999.*

Ng'Ambi, D. 2002. Pre-empting user questions through anticipation: data mining faq lists. In *Proceedings of the 2002 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on Enablement Through Technology*. ACM International Conference Proceeding Series.

Sellberg, L., and Jönsson, A. 2008. Using random indexing to improve singular value decomposition for latent semantic analysis. In *Proceedings of the 6th Conference on Language Resources and Evaluation. Marrakech, Morocco*.

Wang, X., and Jin, X. 2006. Understanding and enhancing the folding-in method in latent semantic indexing. In Bressan, S.; Küng, J.; and Wagner, R., eds., *Database and Expert Systems Applications, 17th International Conference, DEXA 2006, Kraków, Poland, September 4-8, 2006, Proceedings*, volume 4080 of *Lecture Notes in Computer Science*, 104–113. Springer.

Zukerman, I., and Marom, Y. 2006. A comparative study of information-gathering approaches for answering help-desk email inquiries. In *Proceedings of 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia*.

Åberg, J. 2002. *Live Help Systems: An Approach to Intelligent Help for Web Information Systems*. Ph.D. Dissertation, Linköpings universitet, Thesis No 745. http://www.ida.liu.se/~johab/articles/phd.pdf.

Seeing What You Said: How Wizards Use Voice Search Results

Rebecca J. Passonneau[1], Susan L. Epstein[2,3], Joshua B. Gordon[4] and Tiziana Ligorio[2]

[1]Center for Computational Learning Systems, Columbia University
[2]Department of Computer Science, Hunter College of The City University of New York
[3]Department of Computer Science, The Graduate Center of The City University of New York
[4]Department of Computer Science, Columbia University
becky@cs.columbia.edu, susan.epstein@hunter.cuny.edu, joshua@cs.columbia.edu, tligorio@gc.cuny.edu

Abstract

A Wizard-of-Oz experiment investigates how voice search could affect dialogue management strategies. The study design has two novel components. First, a single turn exchange is examined, rather than a full dialogue. Second, wizards partner with a dialogue system, so internal system features unavailable to the wizard can be used to model wizard actions. Wizards see the output of automated speech recognition (ASR) for a book title request, plus a ranked list of candidate titles from a backend query. The features that contribute most to a regression model of the wizards' actions prove to be the utterance level confidence score on the ASR, and the backend return type. People who compare ASR strings to candidate titles can select the correct one if it is there, and do so more confidently when the backend return has higher confidence.

Introduction

For at least the past decade, the quality of automated speech recognition (*ASR*) within spoken dialogue systems (*SDSs*) has been acknowledged as a limiting factor for user satisfaction, task success and other measures of performance (Litman, Walker and Kearns, 1999; Walker et al., 1997). Information-seeking and transaction-based systems (Georgila, et al. 2003, Johnston, et al. 2002, Levin, et al. 2000, Raux, et al. 2006, Zue, et al. 2000) query a backend database for information or to perform actions. The dialogue manager typically maintains system initiative, and aims for short, unambiguous user utterances through carefully designed prompts. This supports maximally accurate backend queries while minimizing clarification subdialogues. CheckItOut, a transaction-based SDS that handles telephone requests for library books, is a mixed initiative system. It accesses a library database where the mean length of the book title field is five words and the median is nineteen. Multiword book titles in the context of book request dialogue acts present an unusual challenge for SDS, particularly with mixed initiative. To address this challenge, we query the backend with ASR for book titles, rather than a semantic interpretation resulting from a natural language understanding phase. This amounts to integrating voice search into SDS.

This paper presents preliminary results of an experiment investigating how voice search could affect dialogue management strategies. The principal findings pertain to three cases of backend return. Humans who compare ASR strings to candidate book titles are justifiably confident in selecting a title when the backend return has high confidence. When the backend has only moderate confidence, our subjects select a title with justifiably less confidence. When the backend return has low confidence, subjects correctly select a title only about a third of the time, and are tentative when they do so.

Voice search has been investigated primarily to access the web via mobile devices (Franz & Milch 2002; Paek & Yu 2008). In our experiment, ASR output is used to query a database of book titles. Often only a few returned titles (*candidates*) will both be roughly the same length as the ASR string and match one or more content words (i.e., nouns, verbs, adverbs and adjectives). For example, for the title *Billy Phelan's Greatest Game,* the ASR output in our experiment was "`billies villains greatest`." A simple query method using that string returned three candidate titles:

- Billy Phelan's Greatest Game
- Baseball's Greatest Games
- More like Us: Making America Great Again

Our subjects' task is to guess which of the candidates returned by the backend query is correct, if any, and to formulate a question if they cannot select a candidate.

The experiment described here relies on the Wizard of Oz (*WOz*) paradigm. In WOz studies, a human subject interacts with a *wizard*, whom she believes to be a computer but is actually a person. Our subjects perform as wizards or as mock callers, using a graphical user interface (*GUI*) rather than a telephone. This work employs two novel adaptations of WOz. First, we examine a single turn exchange, comparable to the well-known notion of adjacency pair (Sacks, Schegloff and Jefferson, 1974), rather than a full dialogue. Second, wizards operate as partners with our dialogue system, which allows us to model their behavior with system-internal features unavailable to wizards, as well as with features representing the wizards' dialogue state.

After an overview of related work, this paper describes CheckItOut and three types of subdialogue likely to arise with voice search around book title requests. Subsequent sections describe the experimental design and results of the experiment, particularly the factors that account for wizards' decisions. The final two sections discuss implications for future work and summarize the contributions presented here.

Related Work

ASR quality, as measured by word error rate (*WER*), typically falls in the range [0.25, 0.65], depending upon such factors as vocabulary size, perplexity of the language model, and diversity of the user population by gender, age, and native language. The impact of WER on SDS performance can also vary considerably, depending on whether the system maintains initiative and on the design of system prompts. CMU's Let's Go!, which provides bus route information to the public from data provided by the Port Authority of Allegheny County, maintains system initiative. The average WER reported in (Raux et al., 2005) was 0.60, due in part to a user population that included elderly and non-native speakers, and in part to the conditions under which users access the system. Callers often called from noisy street locations, or from indoor locations with background noise, such as televisions

Approaches to error-ridden ASR either try to improve the recognizer's data or algorithms, for example through speaker adaptation (Raux, 2004), or try to compensate for transcription errors through error handling dialogue strategies (Bohus, 2004). For the directory service application in (Georgila et al., 2003), users spell the first three letters of surnames, and ASR results are expanded using frequently confused phones. (Stoyanchev and Stent, 2009) add a two-pass recognition architecture to Let's Go! to improve concept recognition in post-confirmation user utterances.

Turn segmentation and disfluencies also affect recognizer performance. A long pause, for example, is likely to be interpreted as the end of the speaker's turn, even if it occurs within the utterance of a long book title. The Let's Go! architecture now has an explicit representation of the conversational floor, the real-time events that take place when speakers seize or cede the next turn (Raux and Eskenazi, 2007). To detect utterance boundaries, an interaction manager uses information from the speech recognizer, a semantic parser, and *Helios*, an utterance-level confidence annotator.

The goal of a WOz study is to elicit behaviors likely to appear when a system replaces the wizard. Work on the impact of ASR errors in full human-wizard dialogues clearly demonstrates that wizards do not aim for full interpretation of every utterance (Rieser, Kruijff-Korbayová and Lemon 2005, Skantze 2003, Williams and Young 2004, Zollo 1999). Zollo collected seven dialogues with different human-wizard pairs whose task was to develop an evacuation plan. The overall WER was 30% and there were 227 cases of incorrect ASR. Nonetheless, wizard utterances indicated a failure to understand for only 35% of them. Instead, wizards ignored words not salient in the domain and hypothesized words based on phonetic similarity. In another study where both users and wizards were treated as subjects, and both knew there was no dialogue system, 44 direction-finding dialogues were collected involving 16 subjects (Skantze, 2003). Despite a WER of 43%, wizard operators signaled misunderstanding only 5% of the time. For the 20% of non-understandings, operators continued a route description, asked a task-related question, or requested a clarification of what had been said.

Simulated ASR controls for the degree of transcription errors, allow collection of dialogues without building or tuning a speech recognizer, and can deliberately deprive the wizard of prosody (Rieser, Kruijff-Korbayová and Lemon, 2005; Williams and Young, 2004). A typist transcribes the user utterances, and errors are introduced systematically. In (Williams and Young, 2004), 144 dialogues were collected simulating tourist requests for information, and WER was constrained to be high, medium or low. High WER decreased full understandings and increased unflagged misunderstandings (where the wizard did not show evidence of detecting the misunderstanding). Under medium WER, a task-related question in response to non-understanding or misunderstanding more often led to full understanding in the next wizard turn than a repair did. Under high WER, when wizards followed a non-understanding or misunderstanding with a task-related question instead of a repair, unflagged misunderstanding significantly increased.

The present experiment is a step towards *wizard ablation*, described in (Levin and Passonneau, 2006), in which the wizard relies on system inputs or outputs, rather than human ones. The hypothesis is that behaviors elicited from wizard/subject pairs in an ablated wizard study will be more pertinent for investigating dialogue strategies given the current state-of-the art in component technologies, such as speech recognition. Here we ablate the input channel to the wizard, so that the wizard has access only to the output of the speech recognizer, not the caller's speech.

In an offline pilot study for this experiment (Passonneau, Epstein and Gordon, 2009), three speakers each read fifty book titles to generate three sets of ASR transcription. Each set was presented to one of three wizards who were asked to find the correct title by searching a plain text file of more than 70,000 titles. WER ranged from 0.69 to 0.83, depending on the speaker. Despite this high WER, on average wizards were able to find the correct title 74% of the time.

The current experiment provides a benchmark for the performance of voice search techniques within the context of CheckItOut, and data on the types of subdialogue to expect for book requests by title. Our initial goals are to identify the contexts in which wizards perform well at selecting the correct title, and especially, to characterize the contexts where they do not, as these are the contexts likely to benefit the most from strategic dialogue management.

CheckItOut

CheckItOut handles book requests made to librarians at the Andrew Heiskell Braille and Talking Book Library. Heiskell is a branch of the New York Public Library and part of the National Library System (*NLS*). Patrons request materials by telephone and receive them by mail. Heiskell and other NLS libraries could greatly benefit from a system that automates some of the borrowing requests.

CheckItOut draws on the Olympus/Ravenclaw architecture and dialogue management framework (Bohus et al., 2007; Bohus and Rudnicky, 2003). *Olympus* is a domain-independent dialogue system architecture based upon the earlier CMU Communicator (Rudniky, 2000). *Ravenclaw* (Bohus, 2004) is a dialogue management framework that separates the domain-dependent task structure from domain-independent error-handling and clarification strategies. Olympus/Ravenclaw has been the basis for about a dozen research dialogue systems in different domains.

CheckItOut has domain-specific code for the task structure of the dialogue. The backend accesses a sanitized version of Heiskell's database of 5028 active patrons, and its full book database with 71,166 titles and 28,031 authors. Titles and author names contribute 54,448 words to the vocabulary.

In a dialogue with CheckItOut, a caller identifies herself, requests books, and is told which are available for immediate shipment and which will go on reserve. The caller can request a book by catalogue number, by title, or by author. We recorded and transcribed 82 calls to the library. Approximately 44% of the book requests were by number, 28% by title or a combination of title and author, and the remainder represented a range of more general book requests. Because patrons receive monthly newsletters listing new titles, they request books with knowledge of the bibliographic data or catalogue numbers. As a result, most title requests from patrons are nearly exact matches to the actual title. For present purposes, we assume they request the exact title or nearly so.

We exploited the Galaxy message passing architecture of Olympus/Ravenclaw to insert a wizard server into CheckItOut. This makes it possible to pass messages from the system to a wizard GUI, or from the wizard GUI to the system. By embedding our wizard within the system, we can examine how wizard actions relate to information available to the system at runtime. Because CheckItOut relies on the same version of Olympus as Let's Go!, we can access features used by the interaction manager mentioned above. This allows us to test whether system features available during the speech recognition phase can be used to model wizards' decisions.

We used PocketSphinx 0.50 for speech recognition, and microphone bandwidth acoustic models from Let's Go!. Like the user population of Let's Go!, patrons of the Andrew Heiskell library include many elderly and non-native speakers. Our target population differs in that patrons qualify for access to Heiskell because they cannot read books in printed format. Many patrons are legally blind, or lack the motor skills to manipulate a book. In separate work, we are evaluating the recognition performance on speech from our transcribed corpus of patron-library calls to determine the utility of additional iterations of acoustic training.

To present challenging cases to our wizards we aimed for a relatively high but not intractable WER. We sought a WER similar to that managed by wizards in the offline pilot study, but with a model that covered the titles in the database. WER was computed using Levenshtein distance (Levenshtein 1996). A statistical language model assigns a probability distribution to possible word sequences. To select a language model, we first manipulated WER by constructing several bigram language models of varying sizes. We randomly selected 10,000 titles (~11K words) from the library database, and then selected from it subsets of size 7,500 (~9K words), 5,000 (~6.8K words) and 1,000 titles (~2K words). For each of the four sets of titles, we constructed a bigram language model. For each language model size, one male and one female each read a set of 50 titles used in our offline pilot. From this, we determined that a language model based on 7,500 titles would yield the desired WER.

To model real-world conditions more closely, titles with below average circulation were eliminated before we selected a set to build the language model for our experiment. We also eliminated one-word titles and those containing non-alphanumeric characters. A random sample of 7,500 was chosen from the remaining 19,708 titles to build a bigram language model. It contained 9,491 unique words. The 4,200 titles in the experimental materials were drawn from the 7,500 titles used in constructing the language model. Average WER for the book title requests in our experiment was 0.69.

Experimental Design

For the current study, we implemented a backend query that returns a ranked list of candidate titles, given the ASR transcription of a caller's book title request. The number of titles in the backend return depends on similarity scores between the ASR string and titles in the database. For the similarity score, we used Ratcliff/Obershelp (*R/O*) pattern recognition, which is the number of matching characters divided by the total number of characters (Ratcliff and Metzener, 1988),. Matching characters are those in the longest common subsequence, then recursively in the longest subsequences in the unmatched regions. For the ASR "`billies villains greatest`" the candidate titles and their R/O scores were:

- Billy Phelan's Greatest Game (0.69)
- Baseball's Greatest Games (0.44)
- More like Us: Making America Great Again (0.44)

Based on our offline pilot, we hypothesized that there would be four distinct cases: a single close match, a small set of competing matches, a larger set of more evenly matched candidates with low but better than random similarity, and no candidates above a low, non-random threshold. The R/O thresholds we selected to yield these four cases here were:

- *Singleton*: a single, good candidate (R/O ≥ 0.85)
- *AmbiguousList*: a list of two to five moderately good candidates (0.85 > R/O ≥ 0.55)
- *NoisyList:* a list of six to ten poor but non-random candidates (0.55 > R/O ≥ 0.40)
- *Empty:* no titles returned at all (R/0 < 0.40)

In each candidate in a list, words that matched a word in the ASR appeared in a darker font, with all other words in grayscale that reflected the degree of character overlap. For AmbiguousList, the darkest font was dark black; for NoisyList it was medium black. Note that our focus here is not on the backend query, but on the distinct types of returns. Certainly, a more finely tuned query could be implemented.

In each *session*, the caller was given a list of 20 titles to read. The acoustic quality of titles read from a list is unlikely to approximate that of a patron asking for a title. Therefore, before each session the caller was asked to read a brief synopsis of each book (taken from the library database) and to number the titles to reflect some logical grouping, such as genre or topic. Titles were then requested in that order.

Participants did two sessions at a time, reversing roles in between. They were asked to maximize a score designed to elicit cooperative behavior and to foster the development of useful strategies. For each individual title request, or *title cycle*, the wizard scored +1 for a correctly identified title, +0.5 for a thoughtful question, and -1 for an incorrect title. The caller received +0.5 for each successfully recognized title. No time limit was imposed on either the session or an individual title cycle. Figure 1 lists the 8 steps in a title cycle.

Seven undergraduate students at Hunter College participated. Two were non-native speakers of English (one Spanish, one Romanian). Each of the 21 pairs of students met for 5 trials. During each trial, one student served as wizard and the other as caller for a session of 20 title cycles, then reversed roles for a second session. The maximum number of title cycles is thus 4,200 (21 pairs × 5 trials × 2 sessions × 20 titles). Participants were allowed to end a session early. We collected data for 4,172 title cycles.

Wizard and caller sat in separate rooms where they could not overhear one another. Each was provided with a headset with microphone, and a GUI. (Audio input on the wizard's headset was disabled.) Both GUIs accepted input from a mouse. The wizard GUI also accepted input from a keyboard.

The wizard GUI presented a live feed of each ASR hypothesis, weighted by grayscale to reflect acoustic confidence. The GUI also included a search field with which to query the database. The wizard selected an ASR string for entry into the search field. Because a long title could be split by the endpointer that segments utterances, wizards could optionally select a sequence of ASR strings. Wizards could also manually edit the search field, but were encouraged not to do so. The search result was presented as a list of candidate titles on the GUI, in descending order of the (unrevealed) similarity score from the backend's retrieval function. Words in returned titles were darkened in proportion to their lexical similarity with the search terms. To offer a title to the caller, the wizard clicked on a title returned by the backend and then on a button labeled "Sure" or "Probably." Selected titles were presented to the caller through a text-to-speech component, prefixed with the word "probably" if the wizard had selected that button. To ask a question instead of selecting a candidate title, the wizard selected two or more titles the question pertained to, clicked a button labeled "Speak" and then spoke into the microphone. Questions could be of arbitrary length and content, and were recorded for offline analysis. The wizard GUI posted the success or failure of each title cycle before the next one began.

1.ASR processes the speech and sends output to the wizard.
2. The wizard can ask the caller to repeat the title one time. The new ASR goes to the wizard.
3. The wizard queries the database either with the ASR string or with words she selects from it.
4. The database backend returns a list of candidates.
5. The wizard selects a candidate with or without high confidence, or selects one or more candidates and asks a thoughtful question intended to help identify the requested title, or gives up.
6. If the wizard selected a candidate, the caller judges its correctness. If the wizard asks a question, the caller judges its reasonableness.
7. The wizard is informed of success or failure, and prompts the caller for the next title.

Figure 1: The title cycle.

The caller GUI gave visual feedback to the caller on the full list of 20 titles to be read during the session. Titles in the list were highlighted green on success, red on failure, yellow if in progress, and not highlighted if still pending. If the caller heard a title selected by the wizard, the caller clicked on "Accept" or "Reject" to rate the wizard's accuracy. If the caller heard the wizard ask a question, the caller clicked on a judgment as to whether she could have answered it ("Can Answer" or "Cannot Answer"). Otherwise the caller clicked to indicate difficulty ("Problem") or uncertainty about the question's relevance ("Undecided").

Evaluation of Wizard and Caller Behavior

Ideally, a wizard should identify the correct title when it is present among the candidates and, if possible, ask a clarifying question when it is not. Our wizards were uniformly very good (95.25% accurate; $\sigma = 1.45$) at detecting a title that was present. They fared less well, however, when the correct title was absent, a situation that occurred 28.36% of the time..

The backend never returned empty on any query, and NoisyLists were rare (2.83%). Responses were nearly evenly divided between a singleton title list (46.74%) and a list greater than one (53.26%). Moreover, every wizard saw a similar distribution of return types from the backend: singleton ($\mu = 278.57$, $\sigma = 21.16$), AmbiguousList ($\mu = 300.57$, $\sigma = 16.92$), and NoisyList ($\mu = 16.86$, $\sigma = 4.78$). The correct title was often (71.31%) in the list of candidates; 92.05% of the Singletons were the correct title, and 53.74% of the AmbiguousLists and NoisyLists contained it.

If the title was present in the backend response, wizards were very good at finding it. When the correct title appeared among the candidates on the wizard GUI (N=2986), the wizard identified it confidently (68.72%) or tentatively (26.53%), a remarkable total of 95.25% of the time. The difficulty of the wizards' task can be evaluated in part by the position of the title read by the caller within the backend response. If the backend returned multiple candidates (N=2222), the first was the correct one 41% of the time. Far less often it was the second (5.81%), third (2.61%), fourth (2.20%), or later. (The fifth through ninth accounted for 1.76%.) This should have helped the wizards, and indeed it did. In those cases where the first on the list was the correct title, wizards offered it 98.34% of the time (74.24% confidently, and 24.10% tentatively).

If the title was not present in the backend response (N=1186), however, wizards performed much less well. After the query return, the wizard was permitted one of four possible actions: *confident* (select a single title with "Sure"), *tentative* (select a single title with "Probably"), *questioning* (ask a question), or *mystified* (the wizard could not formulate a reasonable question and gave up). When the title was not present, the wizards asked a question only 22.32% of the time. Typically our wizards were gamely tentative (67.71%) when the correct title was not among the hypotheses. Less often, they were confident (7.78%) or mystified (2.20%).

One would expect that the way the backend response appeared on the GUI would affect the wizard's action. "Appearance" here refers to the fact that any list was ranked by similarity to the ASR search string, and that words had distinct font color depending on the list type, and the degree of word overlap with the ASR. For each title, we coded the backend response to reflect the likelihood that the return contained the correct title (Singleton = 3, AmbiguousList = 2, and NoisyList = 1), and the wizard's response to reflect her certainty (confident = 3, tentative = 2, questioning = 1, and mystified = 0). The backend response proved somewhat correlated (R=0.59, p < 2.2e-16) with the wizard's response. Although a Singleton (N=1950) from the backend nearly always elicited a title from the wizard (85.38% confident, 13.74% tentative, 0.62% questioning, 0.26% mystified), an AmbiguousList (N=2104) from the backend substantially reduced the wizard's confidence (22.46% confident, 63.28% tentative, 13.32% questioning, 0.95% mystified). The response to NoisyStrings (N=118), was braver than might have been warranted: 9.32% confident, 52.54% tentative, 34.75% questioning, and 3.39% mystified. When the correct title was among the candidates, its *rank* (position in the list of candidates) was somewhat correlated (R=0.42) with the wizard's accuracy (p < 2.2e-16), that is, wizards were more likely to identify a title correctly when it was earlier on the list.

One would also expect wizards' confidence, and therefore their responses, would vary with the individual wizard. Figure 2 confirms this. The ratio of correct decisions to total decisions for each wizard was 0.69 (A), 0.67 (B), 0.66 (C), 0.67 (D), 0.69 (E), 0.69 (F) and 0.70 (G). Over all, the wizards were mostly confident (51.87%) or tentative (40.12%), rarely questioning (7.38%), and almost never mystified (0.62%). Nonetheless, one wizard almost never asked a question, and four did so only rarely. Confidence was correlated with correctness (0.65, p < 2.2e-16). Confident title choices (N=2164) were correct 94.73% of the time; tentative ones only 47.37%. Wizard response type also varied with the caller, as shown in Figure 3. The caller who elicited far more tentative responses and questions than any of the others was the Romanian speaker.

To understand how wizards made *correct* decisions (confident or tentative if the correct title was present, questioning or mystified if it was not), we coded wizards' correctness as correct = 1 and incorrect = 0. A linear regression model was then constructed with 10-fold cross-validation to predict wizard correctness from features available to the wizard or system. Initially we gathered 60 such features, including descriptions of the wizard GUI, how well the ASR matched the candidates and matched database entries, and how well the wizard had done thus far in the current session. Given their interdependence (e.g., different descriptions of the ASR string), preliminary processing examined correlations among the features and reduced the set to 28. The features and the feature selection process are described in detail in (Passonneau et al., Submitted).

The most significant feature in the linear regression model (root relative squared error = 73.60%) was CheckItOut's confidence in its understanding of the caller's reading of the title, which comes from the Helios confidence annotator. While this feature is not available to wizards, it is analogous to how much "sense" the ASR string made to the wizard, and could be used to constrain system behavior. In descending order, the other particularly salient features were the GUI display (Singleton, AmbiguousList, NoisyList), speech rate (faster led to lower accuracy), and on how many of the last three titles the wizard had succeeded. More candidates led to lower accuracy; more words in the ASR string led to higher accuracy. Among the features that made no contribution to the model were the wizard's or the caller's experience at the task (number of sessions to date), and the frequency with which a wizard requested the caller to repeat the title.

Discussion and Future Work

Voice search offered our wizards three types of contexts for book title requests. These translate to three opportunities for CheckItOut. When a single title was returned, wizards justifiably assumed that it was correct. In a full dialogue, CheckItOut could mimic librarians' behavior and simply report the status of the book, without confirming the title with the caller. When an AmbiguousList was returned, wizards made a tentative guess. Half the time, the title was there and the

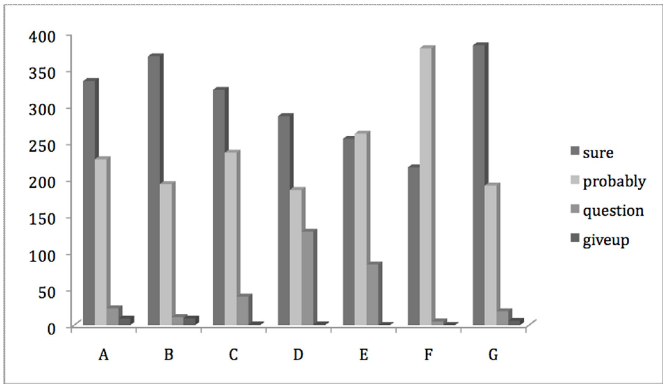


Figure 2: Distribution of acitons chosen by wizard.

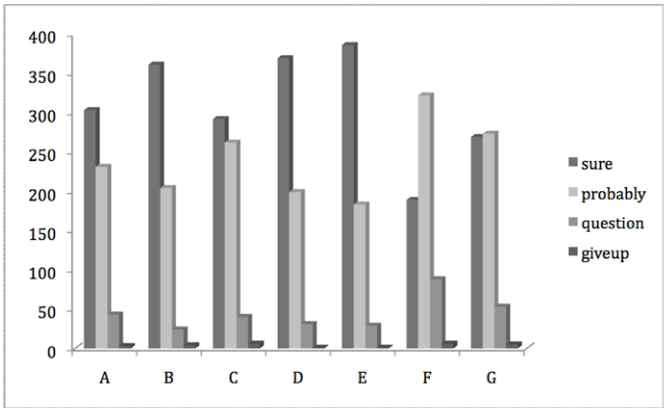


Figure 3: Distribution of actions elicited by caller.

guess was generally correct; the other half, the title was not. Here, CheckItOut could pursue one of two options: implicitly or explicitly confirm a title choice with the caller, or ask a disambiguating question. For example, given the ASR `a charge deaf,` one of our wizards was presented with two candidates: *A Charmed Death* and *A Changed Man*. She then asked "Did you say charmed or changed?" Finally, when the backend returned a NoisyList (six to ten titles), wizards often asked questions about specific words ("*Does it have orchid in it?*"), a strategy bound to be more successful, and appealing to users, than asking for a full repetition.

The focus here has been on the factors wizards attended to when they compared the ASR output to the list of candidates Extensive analysis of individual wizards is the subject of a subsequent paper currently under review (Passonneau et al., Submitted).We logged and computed many more features than those discussed here, including some that gauge the phonetic similarity of the ASR to the title. In addition, wizards and callers completed questionnaires after each session, which we will analyze, along with the wizards' questions, in future work.

Our experiment with voice search extends the WOz paradigm to allow the wizard access only to the ASR of user's utterances rather than to the acoustic input. We have shown that the integration of voice search into dialogue systems has significant promise. The accuracy of the wizards' title offers proved very high. A linear regression model based upon backend return type predicted response type (*confident, tentative, questioning, mystified*) very well. The clear differences in wizard performance bode well for our plans to learn the strategies that make a wizard proficient, and to incorporate those strategies in CheckItOut.

Acknowledgements

This research was supported in part by the National Science Foundation under IIS-084966, IIS-0745369, and IIS-0744904. We thank the staff of the Heiskell Library, our CMU collaborators Alex Rudnicky and Brian Langner and our statistical wizard Liana Epstein. Our undergraduate research assistants provided tireless enthusiasm and painstaking and thoughtful analyses.

References

Bohus, D. 2004. Error Awareness and Recovery in Task-Oriented Spoken Dialogue Systems. Pittsburgh, PA, Carnegie Mellon University.

Bohus, D., A. Raux, T. K. Harris, M. Eskenazi and A. I. Rudnicky 2007. Olympus: an open-source framework for conversational spoken language interface research. *Bridging the Gap: Academic and Industrial Research in Dialog Technology workshop at HLT/NAACL 2007*.

Bohus, D. and A. I. Rudnicky 2003. RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *Eurospeech 2003*.

Georgila, K., K. Sgarbas, A. Tsopanoglou, N. Fakotakis and G. Kokkinakis 2003. A speech-based human-computer interaction system for automating directory assistance services. *International Journal of Speech Technology, Special Issue on Speech and Human-Computer Interaction* 6(2): 145-59.

Litman, D. J., M. A. Walker and M. S. Kearns 1999. Automatic detection of poor speech recognition at the dialogue level. I. *37th Annual ACL*, 309-316.

Passonneau, R., S. L. Epstein and J. B. Gordon 2009. Help Me Understand You: Addressing the Speech Recognition Bottleneck. *AAAI Spring Symposium on Agents that Learn from Human Teachers*, Palo Alto, CA, AAAI.

Passonneau, R., S. L. Epstein, T. Ligorio, J. Gordon, B. and P. Bhutada Submitted. Wizard strategies for resolving noisy ASR against database returns. *10th Annual Meeting on Discourse and Dialogue (SIGDIAL 2009)*.

Ratcliff, J. W. and D. Metzener 1988. *Pattern Matching: The Gestalt Approach, Dr. Dobb's Journal*.

Raux, A. 2004. Automated lexical adaptation and speaker clustering based on pronunciation habits for non-native speech recognition. *Interspeech 2004 (ICSLP)*, Jeju Island, Korea.

Raux, A. and M. Eskenazi 2007. A Multi-layer architecture for semi-synchronous event-driven dialogue management. *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2007)*, Kyoto, Japan.

Raux, A., B. Langner, A. Black and M. Eskenazi 2005. Let's Go Public! Taking a spoken dialog system to the real world. *Interspeech 2005(Eurospeech)*, Lisbon, Portugal.

Rieser, V., I. Kruijff-Korbayová and O. Lemon 2005. A corpus collection and annotation framework for learning multimodal clarification strategies. *Sixth SIGdial Workshop on Discourse and Dialogue*. Lisbon**:** 97-106.

Rudniky, A. I., C. Bennett, et al. 2000. Task and domain specific modeling in the Carnegie Mellon Communicator System. *ICSLP 2000*, Beijing, China.

Sacks, H., E. A. Schegloff and G. Jefferson 1974. A simplest systemics for the organization of turn-taking for conversation. *Language* 50(4): 696-735.

Skantze, G. 2003. Exploring human error handling strategies: Implications for Spoken Dialogue Systems. *ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems*. Chateau-d'Oex-Vaud, Switzerland**:** 71-76.

Stoyanchev, S. and A. Stent 2009. Predicting concept types in user corrections in dialog. *EACL Workshop SRSL 2009*.

Walker, M. A., D. Litman, J., C. A. Kamm and A. Abella 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. *35th Annual ACL*, 271-280.

Williams, J. D. and S. Young 2004. Characterising Task-oriented Dialog using a Simulated ASR Channel. *Eight International Conference on Spoken Language Processing (ICSLP/Interspeech)*. Jeju Island, Korea**:** 185-188.

Subjective, But Not Worthless
- Non-linguistic Features of Chatterbot Evaluations

Pawel Dybala, Michal Ptaszynski, Rafal Rzepka, and Kenji Araki

Graduate School of Information Science and Technology, Hokkaido University
Kita 14 Nishi 9, Kita-ku, 060-0814 Sapporo, Japan
{paweldybala,ptaszynski,kabura,araki}@media.eng.hokudai.ac.jp

Abstract

This paper is a contribution to the heavily neglected field of chatterbot evaluation methods. Based on our experience from research on humor-equipped chatterbots, we propose a methodology for dialogue system evaluation, focusing on their non-task-oriented subtype. Using examples from our previous experiments, we discuss several evaluation methods, discuss what to evaluate, how to do that and how to analyze the results.

1. Introduction

Nowadays, there is a growing need for spoken language dialogue systems (SLDS). To address that need, many research projects are being launched and the field of freely talking computers is gaining more and more interest.

There are two commonly known types of conversational systems: task oriented (such as virtual kiosks or tour guides) and non-task oriented ("chatterbots"). As the area is relatively new, there is an urgent need to work out a robust methodology, also in the field of evaluation. Especially for research on chatterbots the methodology is still quite neglected and there is a lack of even basic standardization.

1.1 Two Areas of Evaluation

Evaluation of dialogue systems depends strongly on their purposes and design. Most existing research projects focus on areas which can generally be divided into two groups:

1) focused on linguistic skills and/or technical quality;

2) focused on non-linguistic skills.

The first area is basically common for both task- and non-task oriented systems. It concerns the system's linguistic skills, such as grammatical correctness, semantic naturalness or vocabulary richness, as well as the technical quality of interaction (response time, voice recognition and generation, visual quality etc.). This area of evaluation is relatively objective.

The second area of evaluation differs for task- and non-task oriented systems. Systems belonging to the former type are designed to achieve specified goals, and this, in most cases, can be seen as the priority in the non-linguistic skills focused aspect of the evaluation. In our opinion, this presence of a specified goal, mutual for user and computer, makes evaluation in this area somehow easier to conduct, as such criteria can be easily and relatively objectively verified.

In the case of chatterbots, however, the non-linguistic skills focused area of evaluation cannot be as easily defined, as there is no mutual goal of the conversation. It is the pleasure of having the interaction that counts. In other words, evaluation of such systems must focus on the user's impressions of the features of the interaction that make it more pleasant, natural and generally "better" in the eyes of the user. Thus, by definition, such assessment has to be subjective.

However, this subjectivity does not necessarily have to be a drawback in chatterbot evaluation. In the end, this is what we want to check – the user's subjective opinion of the product. Another question is how we check it, or – which exact features of the interaction are worth investigating in order to give us the desired results.

1.2 Methodological Gap

Although there are some work reviewing existing methods used to evaluate task-oriented systems (e.g. Dybkjær and Bernsen, 2004), to our knowledge, no robust evaluation methodology for non-task oriented systems has been proposed so far.

In fact, this is the reason why we do not directly compare our methods with other existing research – there is simply nothing we could compare to (we do, however, discuss some particular methods – see below).

In this paper we focus on the non-linguistic area of chatterbot evaluation. However, the methods described here can also be applied in experiments on task-oriented systems, with some slight changes in their content.

2. Systems Used in This Research

The methods and perspectives of chatterbot evaluation presented in this paper have been used in our research on humor-equipped talking systems. In this section we briefly describe two chatterbots used in the research along with an emotiveness analysis system, used in the automatic evaluation experiments (see **4.3**).

2.1 Two chatterbots

The baseline system in our research is a Japanese, text based chatterbot called "Modalin" (developed by Higuchi et al, 2008). The system extracts keywords from user utterances, uses them to extract word associations from the Internet and adds modality to the generated responses (for details, see Higuchi et al. 2008).

The chatterbot was used as a base to create a Japanese pun-telling system "Pundalin". To do that, we added a pun generating engine developed in our previous research (Dybala et al., 2008) to Modalin, using a very simple timing rule - in every third turn of the conversation, the system's output was replaced by a joke-including sentence, generated by the joking system (for details, see Dybala et al., 2008). Currently we are working on a more sophisticated algorithm.

2.2 ML-Ask System

In our research we used the ML-Ask Emotive Elements/Emotive Expressions Analysis System for Japanese (Ptaszynski et al., 2008) to perform automatic analysis of chat logs acquired in the user-focused experiment (see **4.3**). The ML-Ask system performs utterance analysis in two general steps:

1. **Determining general emotiveness (emotive/non-emotive), and**
2. **Specifying valence and types of emotions found (positive/negative plus specified type).**

In the second step, analysis of the specific emotions showed by the evaluators during the conversation provided us with the information of their feelings towards the system.

In our research, we separated the emotion types and emotion expressions using two general dimensions: positive/negative and activated/deactivated (Russel, 1980). Each type of emotion can be described using these two dimensions.

The types of emotions in our research are based on Nakamura's (1993) Japanese emotion classification (10 types). The proposed emotion types were projected on Russel's 2-dimensional model of affect (Russel, 1980). The effect of this projection can be seen in Figure 1.

3. Our Evaluation Methodology

The subjectivity of non-linguistic features of chatterbots evaluation and the need for "measuring" user impressions of the system do not give us much of a choice when conducting experiments. The easiest and most obvious method to do that is to ask users directly what they think of the interaction.

Although of high importance, evaluation conducted by users also has its drawbacks (discussed below). Thus, in our research we employed two complementary, non-user focused evaluation methods: third person focused evaluation and automatic (emotiveness analysis based) evaluation.

Figure 2 presents the outline of the methodology proposed in this paper. Needless to say, it still needs to be improved – we hope this paper will trigger a thorough discussion on this subject.

Below we describe and discuss the methods from the non-linguistic area, using examples from our earlier research. In this paper we focus on non-task oriented systems.

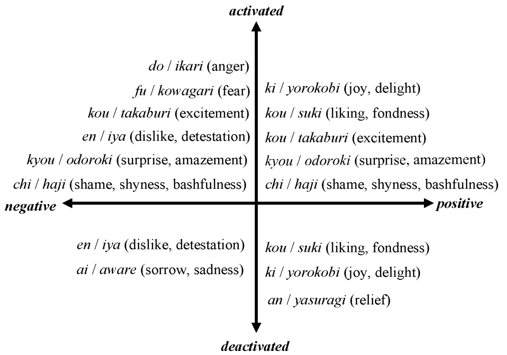

activated
do / ikari (anger)
fu / kowagari (fear)
kou / takaburi (excitement)
en / iya (dislike, detestation)
kyou / odoroki (surprise, amazement)
chi / haji (shame, shyness, bashfulness)
ki / yorokobi (joy, delight)
kou / suki (liking, fondness)
kou / takaburi (excitement)
kyou / odoroki (surprise, amazement)
chi / haji (shame, shyness, bashfulness)
negative
positive
en / iya (dislike, detestation)
ai / aware (sorrow, sadness)
kou / suki (liking, fondness)
ki / yorokobi (joy, delight)
an / yasuragi (relief)
deactivated

Figure 1 Grouping Nakamura's classification of types of emotions on Russell's two-dimensional space

4. What to Evaluate and How?

This may sound completely obvious, but we need to state one thing before we start explaining and discussing our evaluation methods: the shape of the evaluation experiment strongly depends on what we want to check. In our research we focus on non-task oriented humor-equipped dialogue systems. In the evaluation experiment we explore both the linguistic- and non-linguistic area – however, we focus on the latter, as it is the role of humor in the interaction we want to check in the first place. For the same reason, some of the questions in our evaluation were directly related to humor and perceived funniness of the systems' utterances.

Taking these two aspects into consideration, in our evaluation we decided to explore such non-linguistic aspects of interaction as: human-likeness, the will to continue the dialogue, engagement in the conversation, funniness and emotive response. We also evaluated the linguistic area of the systems' performance – however, in our research these results were of secondary importance.

These issues are described in details below.

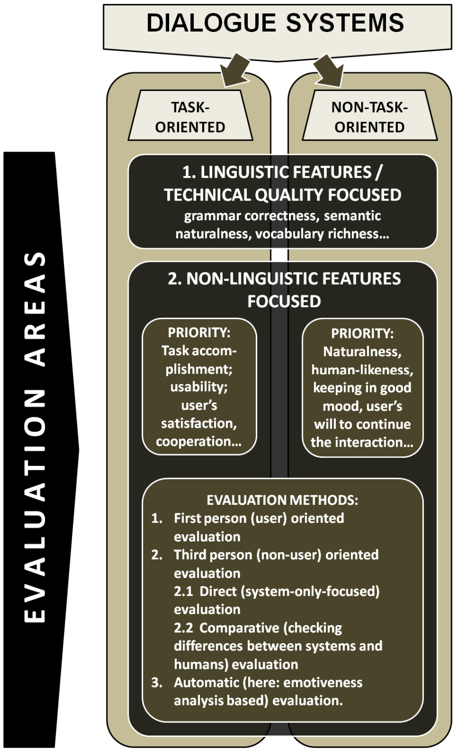

DIALOGUE SYSTEMS
TASK-ORIENTED
NON-TASK-ORIENTED
EVALUATION AREAS
1. LINGUISTIC FEATURES / TECHNICAL QUALITY FOCUSED
grammar correctness, semantic naturalness, vocabulary richness...
2. NON-LINGUISTIC FEATURES FOCUSED
PRIORITY: Task accom-plishment; usability; user's satisfaction, cooperation...
PRIORITY: Naturalness, human-likeness, keeping in good mood, user's will to continue the interaction...
EVALUATION METHODS:
1. First person (user) oriented evaluation
2. Third person (non-user) oriented evaluation
2.1 Direct (system-only-focused) evaluation
2.2 Comparative (checking differences between systems and humans) evaluation
3. Automatic (here: emotiveness analysis based) evaluation.

Figure 2 Evaluation areas of task- and non-task oriented dialogue systems

4.1 First Person Oriented Evaluation

As it is users who are the “clients” of our product, in the research on dialogue systems the first person oriented evaluation is to us of highest importance. Although not perfect, methods described in this section allow us to check the user impressions of the interaction with the system, in possibly the most direct way. As stated above, this evaluation is by definition subjective, but we do not see it as its drawback. Instead, we propose to accept the lack of objectivity as a natural fact in impression-relying evaluation. Individual differences are inevitable here, and their presence can be a trigger to construct more sophisticated, user-adapting systems. An idea of such a system for humor-oriented chatterbots is described in one of our earlier works (Dybala et al., 2009).

There are two major methods of first person oriented evaluation: one in which user impressions of the interaction are checked by filling out a questionnaire or conducting an interview, and another one conducted using the Turing Test (Turing, 1950). The latter has for years been the subject of many discussions and actually received a lot of criticism. Therefore, as this method is probably the best known of all methods mentioned in this paper, we are not going to discuss it in detail here. We would only like to mention that – comparing to the questionnaire/interview method – the Turing Test does not give us detailed and measurable information about the interaction, since its only aim is to check if the system is sophisticated enough to deceive users about it not being human.

Therefore, in our research we used the former first person evaluation method to conduct the first person oriented evaluation experiment. Human evaluators were asked to perform a 10-turn dialogue with Modalin (non-humor-equipped), and then with Pundalin (humor-equipped system). No topic restrictions were made.

There were 13 participants in the experiment, 11 male and 2 female; all of them were university undergraduate students. After talking with both systems, they were asked to fill out a questionnaire about each system’s performance. The questions concerned both linguistic (B-D) and non-linguistic (A, E-H) areas of interaction:

A) Do you want to continue the dialogue with the system?; **B)** Was the system’s output grammatically natural?; **C)** Was the system’s output semantically natural?; **D)** Was the system’s vocabulary rich?; **E)** Did you get an impression that the system possesses any knowledge?; **F)** Did you get an impression that the system was human-like?; **G)** Do you think the system tried to make the dialogue more interesting?; **H)** Did you find the conversation with the system interesting?

The replies to the questions were given on 5-point scales with some explanations added. Each evaluator filled out two such questionnaires, one for each system. The final, summarizing question was “Which system do you think was better?”. Statistical significance of the results was calculated using the Student’s t-test. The results are summarized in Table 1.

Question	**Modalin**	**Pundalin**	**Difference**	**P value**
A	2.62	3.38	**0.76**	>0.05
B	2.15	2.92	**0.77**	>0.05
C	1.85	2.69	**0.84**	**<0.05**
D	2.08	3.00	**0.92**	**<0.05**
E	2.15	2.85	**0.70**	**<0.05**
F	2.38	3.31	**0.93**	**<0.05**
G	1.92	4.15	**2.23**	**<0.05**
H	2.46	4.08	**1.62**	**<0.05**
Which is better?	15%	85%		

Table 1 User evaluation – results for Modalin and Pundalin for detailed questions. Answers were given on a 5-point scale.

The results show that the system with humor received higher scores in both linguistic and non-linguistic areas. As for the former, it may seem unusual that the presence of humor improved the system’s linguistic skills – this fact, however, could have been caused by the fact that Pundalin uses fragments of human created sentences and jokes from a data base, which naturally are more correct than those automatically generated by the computer.

Also in the non-linguistic area all results point at the humor-equipped system. Users wanted to continue the conversation to a higher degree with Pundalin than with Modalin, perceived Pundalin as more human-like, knowledgeable, funny and generally better than Modalin (Dybala et al., 2008).

Results for questions A and B were found to be significant on the 6% level, and for remaining questions – on the 5% level.

Discussion

As mentioned above, the first person oriented method is the best and most direct way of evaluating the system. Such a method (questionnaire/interview) was also used by Bernsen and Dybkjær in their experiments on NICE - a system for spoken and gesture interaction with life-like fairytale author Hans Christian Andersen (Bernsen and Dybkjær, 2004). The content of the questions was slightly different from those we used, as the embodiment of the system and the usage of gestures also needed to be addressed. However, the general tendency in NICE’s evaluation was consistent with ours: most questions concerned the users’ subjective impressions of the interaction (such as: “How did it feel to talk to the system?”). The biggest difference between our and Bernsen and Dybkjær’s evaluation was qualitative – e. g., the answers in their experiment were given freely by the users, without any quantitative scale. User responses were manually analyzed and their descriptive summarization is the result of the experiment. Such non-quantitative methods surely can give us a deeper insight into the users’ impressions of the system – however, it is quite hard to use when comparing with other systems. Therefore, we think that the qualitative methods should be used in preliminary experiments, in order to receive feedback from the users, rather than to evaluate the final product.

While of high importance, user-focused evaluation also has its drawbacks. First, even when conducted immediately after the interaction, it requires the user to remember his/her impressions from when the conversation took place. On the other hand, the alternative here would be asking the user to evaluate the system during the conversation, which might distract him/her and negatively influence the smoothness of the interaction. One way to solve this problem is to also conduct a third person oriented experiment (see **4.2**).

Another problematic issue of the first person oriented evaluation method is that we cannot be sure if the user is actually aware of his/her own feelings and emotions. In our research we partially solved this problem by using the automatic emotiveness analysis based evaluation. Even if the users do not fully realize their feelings and do not reflect them in the evaluation, there is a chance that they will be reflected in what they say. Therefore, emotiveness analysis of the user utterances can be seen as a good complementary evaluation method.

4.2 Third Person Oriented Evaluation

In the first person oriented evaluation we ask the users directly about their impressions of the system. However, as mentioned above, one drawback of this method is that the evaluation has to be conducted after the conversation. To solve this problem and double-check the results of the first experiment, we conducted an additional experiment, in which third person (non-user) participants evaluated the chat logs from the user experiment. The questions asked were similar to those used in the user-focused experiment – we only made minor adjustments. First, the word "system" was changed to "dialogue" (in some cases - "Speaker"), as we did not want the evaluators to know that some of the utterances were generated by a computer system. In the chat logs given to the third person evaluators, dialogue participants were called "Speaker A" for the user and "Speaker B" for the system. In addition, question F (about human-likeness) was deleted, as it would also reveal that at least one speaker was not human. Also, in questions B, C, D, E, G and H we added two options: 1) "Speaker A" and 2) "Speaker B" – so that the dialogue participants would be evaluated separately. Thus, the list of questions used in this experiment goes as follows:

> **A)** Do you want to read the continuation of the dialogue?; **B)** Was Speaker A/B's talk grammatically natural?; **C)** Was Speaker A/B's talk semantically natural?; **D)** Was Speaker A/B's talk vocabulary rich?; **E)** Did you get an impression that Speaker A/B possesses any knowledge?; **F)** <deleted>; **G)** Do you think the Speaker A/B tried to make the dialogue more funny and interesting?; **H-1)** Did you find the dialogue interesting in general? **H-2)** Did you find Speaker A/B's talk interesting?

After completing the detailed questionnaire, the evaluators answered the final question, the same as in the previous experiment - "Which dialogue did you find most interesting?" (we used the Japanese word *omoshiroi*, which can mean "interesting" or "funny", and is generally positive in meaning (Nakamura, 1993)).

The chat logs were divided into 13 sets. Each of them included one Modalin and one Pundalin dialogue. Each set was evaluated by 5 participants, which makes a total of 65 evaluators, all of which were university students. (Dybala et al., 2008). Statistical significance of the results was calculated using the Student's t-test.

The results were analyzed using two methods mentioned above: direct and comparative.

4.2.1 Direct Evaluation

In this method we only take into consideration the results for systems' (Speaker B's) utterances, and compare them for both humor- and non-humor equipped system, as we did in first person evaluation. The results of this method are summarized in Table 2.

Although the differences here were not that clear and significant as in the user-focused evaluation, the tendency is still visible. The humor-equipped system received higher scores in all categories. Only in two cases (D and F) were the differences found to be statistically significant – however, the results for the general question show that even if there is not much difference, the evaluators still chose dialogues with humor (69% vs. 31%).

Question	Modalin	Pundalin	Difference	P value
A	2.60	2.89	**0.29**	>0.05
B	1.78	2.09	**0.31**	>0.05
C	1.48	1.69	**0.21**	>0.05
D	2.03	2.38	**0.35**	**<0.05**
E	1.87	2.13	**0.26**	>0.05
F	X	X	X	X
G	2.51	2.91	**0.40**	**<0.05**
H-1	2.88	3.19	**0.31**	>0.05
H-2	2.73	3.16	**0.43**	>0.05
Which is better?	31%	69%		

Table 2 Third person evaluation – results for Modalin (non-humor equipped system) and Pundalin (humor-equipped system). Answers were given on a 5-point scale.

Discussion

As shown in the Table 2, the results are generally consistent with those of the first person oriented experiment. Thus, it can be stated that the direct third person oriented method can be used to evaluate chatterbot performance. However, the low differences and lack of significance in this experiment require some detailed discussion.

One possible explanation of this phenomenon is that when users are talking with the systems, they are usually quite impressed by the very fact that a computer can talk. This very fact may positively influence the results. With this understanding, the third person oriented evaluation seems more objective, since the evaluators were not participants of the interaction, and thus had more distance to the subject of evaluation. Also the fact that they did not know that one of the speakers was a computer system was not without meaning.

Albeit the relative "objectiveness" of the third person evaluation (more distance towards chat logs than towards conversation partner), this method has a few drawbacks. The major one is that, as mentioned above, it is the user that has to be satisfied in the first place. They will use the system and it is their opinion that counts the most. Of course, the more severe and diversified the evaluation, the more information about our systems and enhancements needed we can get; however, to evaluate the final product it is still the first person oriented evaluation that should be of primary importance, and third person oriented methods should be used rather to double-check the results or to acquire feedback leading to the system's development.

4.2.2 Comparative Evaluation

As mentioned above, in our third person evaluation experiment we referred to the speakers in the chat logs as Speaker A (the user) and Speaker B (the system). In the direct evaluation we took into consideration only the results for Speaker B, while in the comparative evaluation we calculated the differences between the systems and the users. Statistical significance of all scores was calculated using the Student's t-test. The results are summarized in Tables 3 and 4.

Modalin				
Question	**User**	**System**	**Diff.**	**P value**
B	3.30	1.78	**1.52**	<0.05
C	2.94	1.48	**1.46**	<0.05
D	2.92	2.03	**0.89**	<0.05
E	3.13	1.87	**1.26**	<0.05
G	2.54	2.51	**0.03**	>0.05
H-2	2.85	2.73	**0.12**	>0.05

Table 3 Results for Modalin for detailed questions in third person evaluation (differences between users and systems). Negative values mean that Speaker B (the system) received higher scores than the user.

Pundalin				
Question	**User**	**System**	**Diff.**	**P value**
B	3.18	2.09	**1.09**	<0.05
C	3.00	1.69	**1.31**	<0.05
D	2.81	2.38	**0.43**	<0.05
E	2.97	2.13	**0.84**	<0.05
G	2.52	2.91	**-0.39**	<0.05
H-2	3.09	3.16	**-0.07**	>0.05

Table 4 Results for Pundalin for detailed questions in third person evaluation (differences between users and systems). Negative values mean that Speaker B (the system) received higher scores than the user.

As shown in the above tables, the results show that the humor-equipped system differs less from humans than the non-humorous one. In other words, the difference between humans and Pundalin was smaller than the difference between humans and Modalin. In our research this is especially important for questions D-H, which belong to the non-linguistic area of evaluation. Looking at the results, we can see that the system with humor actually made more effort than the users to make the dialogue interesting. The fact that Pundalin surpassed the users in this category can be interpreted as not necessarily positive, as trying too hard may also be annoying. However, knowing that the system was assessed as generally better both by the users and third person evaluators, we can assume that the attempts to make the conversation more interesting were rather appreciated than disliked.

Discussion

From these results, a conclusion can be drawn that the system which differs less from humans can be seen as more human like. This assumption is consistent with the results of the first person oriented experiment (see **4.1**, question F). Obviously, more research on the issue of human-likeness is needed – however, we think that the method suggested here is also an option for checking how close to the human level the system is.

This method, albeit innovative, has also several drawbacks. The main one is the same as in case of the direct evaluation (see **4.2.1**) – it may be slightly less subjective, but it is also less direct, and does not involve the users. However, the consistency with the results of the user oriented experiment shows that it can be used as a complementary method of evaluation.

In previous sections we mentioned the Turing Test, as probably the best known (although arguable) method for checking the system's human likeness. The Turing test is a first person oriented method, in which the users have to tell if the interlocutor is a human or a computer. However, it should be possible to conduct a third person oriented version, in which the evaluators would read the chat logs and guess the identity of speakers. Obviously, third person oriented Turing Test would have the same drawback as other third person oriented methods – however, we believe that it can be a good complementary method and as such may be worth trying.

4.3 Automatic Evaluation

In the above sections we discussed some drawbacks of first and third person oriented evaluation methods (see **4.1** and **4.2**). These problems can be solved by using automatic methods, in which the system's performance is evaluated by another system. In our research we used the ML-Ask System (see **2.2**) to perform automatic analysis of chat logs acquired in the user-focused experiment.

4.3.1 General Emotiveness

In the first step, general emotiveness (emotive/non-emotive) of user utterances was analyzed by the ML-Ask system. Most of the users (11 out of 13) showed more emotions towards Pundalin than towards Modalin, which means that they were generally more emotively engaged in the conversation with the system which used humor (Ptaszynski et al., 2008).

4.3.2 Valence and specification

In the next step of the evaluation, the chat logs were analyzed to check the specific types of emotions in the users' utterances. Figure 4 shows the results projected on the Russel's two-dimensional space (see also **2.3**, Figure 1).

As shown above, while most of emotions towards Modalin were negative and activated (45%, 78% of negative emotions in total), for Pundalin the proportion

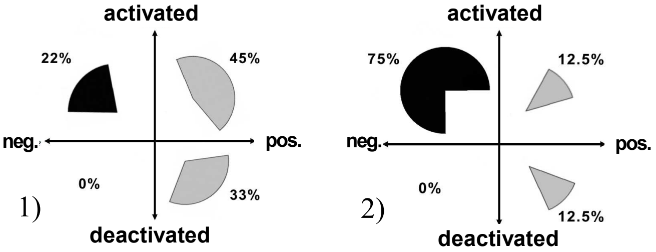


Figure 3: Projection of emotive analysis of users' emotions types on Russell's two-dimensional space – 1) Modalin (without humor) and 2) Pundalin (with humor)

was opposite (75% of positive and activated). In this experiment, no positive deactivated emotions were found in either the humor- nor non-humor-equipped system's chat logs.

4.3.3 Positive/Negative Engagement

The correlation between the speaker's emotiveness and conversation engagement has been shown in various research (e.g. Goodwin and Goodwin, 2000). This knowledge was also used by Yu et al. (2004) in their research, in which they measured engagement level based on emotions recognized in user speech. The approach is quite similar to the one presented in this section – however, in our research, we used emotions extracted from the textual layer of the conversation, and focused on the influence of humor as a measure to improve the engagement.

Conclusions about the users' engagement in the conversation were drawn based on the results described in **4.3.1** and **4.3.2**. General emotiveness analysis results (see Figure 3) suggest that the users were more engaged in the conversation with Pundalin, as they showed more emotions towards it than towards Modalin. However, the engagement does not necessarily have to be positive, as the users might have reacted to the systems' utterances with negative arousal or irritation. This is why we propose to distinguish between positive and negative engagement. To do that, in our research we use emotive valence (positive/negative emotions – see Figure 3). The results show that the users were generally more positively and less negatively engaged in the conversation with Pundalin. This is consistent with the results of the previous experiments, especially with the questions concerning the will to continue the interaction (see **4.1**). Therefore, it can be stated that automatic emotiveness analysis can also be used to investigate user engagement and its types.

The role of activation/deactivation of emotions is still to be investigated.

Discussion

In this section we described another automatic evaluation method used in our research. This method has a few significant advantages. First: it is automatic. This means that it does not require any additional engagement from the users in the experiment – they only have to perform a conversation with the system. We do not have to waste our or other people's time. The evaluation can be conducted at any time, not necessarily right after the conversation.

Second: as mentioned above, it is quite difficult to speak about one's own emotions and feelings, as we may not be fully aware of all of them. Not mentioning the fact that – if the evaluation is conducted after the conversation – users may not exactly remember what they felt some time ago. These feelings and emotions, however, can be revealed in the users behaviour during the interaction, and also in the textual layer, which, in case of our text-based-only system, is of high importance.

Of course, even during conversation with text-based systems like ours, emotions can also be detected from other layers than the textual. Such methods as voice or facial expression recognition-based emotions detection should also be taken into consideration (and, possibly, combined with the textual-based one we used).

5. Conclusions

In this paper we proposed a methodology for chatterbot evaluation, focusing on the non-linguistic area. The proposed methods have been experimentally tested, so it can be stated that they are at least applicable and can be used also in other research on non-task oriented dialogue systems.

Obviously, the methods and methodology are not perfect and there are many more issues that have to be taken into consideration, such as system embodiment, voice recognition/generation or user-system interaction length and frequency.

Acknowledgments

This work was partially supported by a Research Grant from the Nissan Science Foundation and Hokkaido University Global Center of Excellence (GCOE).

References

Bernsen, N.O., and Dybkjær, L. 2004. Evaluation of spoken multimodal conversation. In the Proc. of ICMI'04. 38-45. New York: ACM Press.

Dybala, P., Ptaszynski, M., Higuchi, S., Rzepka, R., and Araki, K. 2008. Humor Prevails! - Implementing a Joke Generator into a Conversational System. In the Proc. of AI-08, Wobcke, W. and Zhang, M. (eds), Auckland, New Zealand, 2008. Springer-Verlag LNAI Vol. 5360, 214-225, Springer Berlin & Heidelberg.

Dybala, P., Ptaszynski, M., Rzepka, R., and Araki, K. 2009. Humorized Computational Intelligence - towards User-Adapted Systems with a Sense of Humor. In the Proc. of the EvoWorkshops 2009. M. Giacobini et al. (Eds.). Springer-Verlag LNCS, Vol. 5484, pp. 452–461, Springer Berlin & Heidelberg.

Dybkjær, L., Bernsen, N. O., and Minker, W. 2004. Evaluation and usability of multimodal spoken language dialogue systems. *Speech Communication 43*(1–2): 33–54.

Goodwin, M.H., and Goodwin, C. 2000. Emotion within situated activity. *Communication: An Arena of Development*. 33–54. N. Budwig, I. C. Uzgiris, J. V. Wertsch, Eds. Ablex.

Higuchi, S., Rzepka, R., and Araki, K. 2008. A Casual Conversation System Using Modality and Word Associations Retrieved from the Web. In the Proc. of EMNLP'08. 382-390. Honolulu, USA.

Nakamura, A. 1993. Kanjo hyogen jiten. (Dictionary of Emotive Expressions) (in Japanese). Tokyodo Publishing, Tokyo, Japan.

Ptaszynski, M., Dybala, P., Higuchi, S., Rzepka, R., and Araki, K. 2008. Affect as Information about Users' Attitudes to Conversational Agents. In the Proc. of HAI'08. 459-500, Sydney, Australia.

Russell, J. A. 1980, A circumplex model of affect. *Journal of Personality and Social Psychology, 39*(6):1161-1178.

Turing, A. 1950. Computing Machinery and Intelligence. *Mind 59*(236), 433-460.

Yu, C., Aoki, P.M., and Woodruff, A. 2004. Detecting user engagement in everyday conversations. In the Proc. of ICSLP 2004. 1329-1332. Jeju Island, Korea.

SpeechEval – Evaluating Spoken Dialog Systems by User Simulation

Tatjana Scheffler and **Roland Roller** and **Norbert Reithinger**
DFKI GmbH
Alt-Moabit 91c
10559 Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

In this paper, we introduce the SpeechEval system, a platform for the automatic evaluation of spoken dialog systems on the basis of learned user strategies. The increasing number of spoken dialog systems calls for efficient approaches for their development and testing. The goal of SpeechEval is the minimization of hand-crafted resources to maximize the portability of this evaluation environment across spoken dialog systems and domains. In this paper we discuss the architecture of SpeechEval, as well as the user simulation technique which allows us to learn general user strategies from a new corpus. We present this corpus, the VOICE Awards human-machine dialog corpus, and show how this corpus is used to semi-automatically extract the resources and knowledge bases on which SpeechEval is based.

1. Introduction

The more spoken dialog systems (SDSs) are put into practice in different domains, the more efficient methods for their development and deployment are urgently needed. The project SpeechEval aims to address this need in two ways: First, by investigating the use of dialog corpora in order to automatically or semi-automatically create the resources necessary for the construction of SDSs. And second, by learning general user behavior from the same corpora, and building a flexible user simulation which can be used to test the overall usability of SDSs during development or after deployment.

Automatic testing of dialog systems is attractive because of its efficiency and cost-effectiveness. However, previous work in this area concentrated on detailed tests of individual subcomponents of the SDS (such as the ASR), or on small systems in toy domains. In order to judge the overall usability of a commercial dialog system, extended testing by human callers has been necessary – a step that is usually too costly to be undertaken during the prototype stage or repeatedly after changes to the deployed system. SpeechEval intends to fill this gap, providing a flexible user simulation platform which allows automatic repeated testing of an SDS. Maximum modularity of the system architecture as well as the automatic and semi-automatic techniques for the creation of the underlying resources for the user simulation (in particular, domain knowledge and user strategies) allow SpeechEval to be easily portable across different SDSs.

In this paper, we concentrate first on the user simulation technique in SpeechEval. Then we describe the architecture of the SpeechEval platform. We pay special attention to the resources (general, domain- or system-dependent) which need to be constructed or adapted when using SpeechEval as a user simulation for a new application. The rest of the paper describes our finished and ongoing work in extracting knowledge bases for the SpeechEval system from corpora.

2. User Simulation

User simulation is used in the SDS literature for several purposes. First, for training the dialog manager of a spoken dialog system during reinforcement learning. In this case, the SDS with the learned strategy is the actual purpose of the research, whereas the user simulation is just a means to that end. Second, user simulation is used for evaluation or testing of the trained policies/dialog managers of the developed spoken dialog systems. The two types of purposes of user simulations may call for different methods. A good overview of state-of-the-art user models for SDS training is given in (Schatzmann et al. 2006). A user simulation may be used to test for general soundness of an SDS, specifically searching for errors in the design. In such a case, a random exploration may be called for (Alexandersson and Heisterkamp 2000). A restricted random model may also perform well for learning (Ai, Litman, and Litman 2007).

In other cases, ideal users may be modelled so that reinforcement learning is able to learn good paths through the system's states to the goal (López-Cózar et al. 2003). In an approach closer to our work, (Chung 2004) developed a variable user simulation used for detecting potential errors in a SDS with a large database back-end. In both projects, the user simulation is hand-crafted by the designer of the SDS.

Our goal in SpeechEval is to as much as possible avoid hand-crafting the strategy (i.e., user simulation). Since in our case the user simulation itself is the goal and not merely a step along the way, the requirements for the user model may also differ from previous approaches. An optimal strategy is not needed for our user simulation, neither is a random explorative strategy. Instead, the aim should be *realistic* user behavior. Since SpeechEval should be used to evaluate spoken dialog systems in parallel or instead of human judges, it should show similar behavior (at least asymptotically) to these judges. The behavior of human evaluators of spoken dialog systems can be observed in our corpus, the VOICE Awards Corpus described below in section 4. We therefore define realistic user behavior in our case as user utterances that probabilistically match the ones represented in our corpus. Such probabilistic models are often used for evaluation of learned dialog managers (Ai, Litman, and Litman 2007). How to effectively measure the realism of simulated dialogs is still very much an open research question. Some measures are discussed for example in (Jung et al. 2009), based on comparing the simulated dialogs with real user dialogs using the BLEU metric and based on human judgments. In the absence of real user dialogs with the same SDS, we aim for greater variability in the simulated user behavior.

One method of achieving both greater realism and variability is the use of a true speech interface when interacting with the SDS to be evaluated. Previous work often reduces interaction to the text or even concept level, or uses canned user responses (as in the case of (López-Cózar et al. 2003)). In contrast, SpeechEval interacts with the SDS just like a human user would, over the telephone. The use of a text-to-speech system allows for greater variability in production than concept-based or canned output. It will allow us to tune the output and introduce disfluencies as well as errors and uncooperative behavior. On the other hand, using ASR and TTS modules obliviates the need to artificially "model" signal errors by introducing fixed error rates. Instead, errors will be introduced naturally through the normal telephone noise. The ASR component shows very good results so far, which should be able to match a human user as long as the ASR grammar is suitable. Furthermore, robust processing in the pipeline ensures that small ASR errors will not completely derail the response. Overall, the use of a real speech interface makes the simulated dialogs much more realistic and variable than it would otherwise be possible.

3. SpeechEval Architecture

The planned architecture of the SpeechEval system is shown in Figure 1. It essentially follows a standard pipelined architecture for spoken dialog systems, with some additional modifications to include the user simulation functionality. In this section, we briefly describe the components of our system, and the resources which are necessary to use SpeechEval to evaluate a given SDS. Such resources may be general, domain- or system-specific. We discuss in each case, whether they must be specified by hand or can be learned (and how).

SpeechEval will be implemented using the Ontology-based Dialogue Platform (ODP) a generic modeling framework and run-time environment for multimodal dialog applications. For a more detailed description, see (Pfalzgraf et al. 2008).

There are three central knowledge bases which need to be acquired off-line before launching the system: (1) A domain ontology, which contains domain-specific information about available objects and actions and must be specified by hand. SpeechEval provides functionality which supports and speeds up the construction of this ontology. (2) A set of user goals to be used during the user simulation. Such goals are highly system and domain specific and must be specified by a domain expert. This goal set is equivalent to the instructions provided to human testers and therefore does not in itself constitute a significant impediment of using SpeechEval for automatic testing. (3) A user stereotype. Possible user characteristics are extracted from a dialog corpus (see below). SpeechEval allows testing with different user characteristics (such as fast or slow reaction time, many/few barge-ins, or differing error rates). A GUI is planned which allows the SpeechEval user to set these characteristics in an intuitive way.

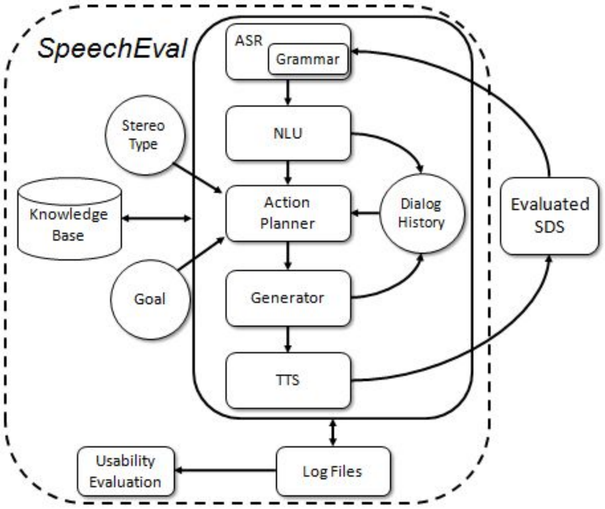


Figure 1: Architecture of the SpeechEval system.

During on-line runs, SpeechEval's architecture largely follows a standard pipeline model. The speech signal with the SDS prompt received via telephone is first processed in the ASR component. The recognition grammar is learned in a multi-step process using our human-machine dialog corpus (introduced in section 4) as well as other sources. This obliviates the need for tedious hand-tuning of the grammar, and thus makes SpeechEval much more easily portable to new SDSs and domains.

The second step of natural language understanding (NLU) consists of three parts. The segmentation and dialog act classification components are learned from our annotated corpus. We follow the approach in the AMIDA project (AMIDA 2007) for the segmentation. For the dialog act classification, we use a Maximum Entropy classifier trained with the WEKA toolkit (Witten and Frank 2005). Our implementation is based on the work by (Germesin 2008). However, in an on-line system such as SpeechEval, features based on future dialog act assignments cannot be used. The third component of the NLU module performs a keyword search and other information retrieval steps to link the incoming prompt to the domain ontology.

The action planner is the central step in the pipeline. Based on the analysis of the incoming prompt, a reply action is devised. Our current target approach is very close to the one proposed in (Georgila, Henderson, and Lemon 2005) for an information state update system. At each state in the dialog, the user model choses the next action based on the transition probabilities observed in the corpus. Since some states have never or only rarely been seen in the corpus, we choose a vector of features as the representation of each dialog state. These features in our case include properties of the dialog history (such as the previous dialog act, the number of errors), the current user characteristics (expert vs. novice, for example), as well as other features such as the ASR confidence score. We estimate from the corpus the amount that each feature in the vector contributes to the choice of the next action. Thus, unseen states can be easily mapped onto the known state space as they lead to similar behavior as closely related seen states would.

The chosen next action is a dialog act type that must be enriched with content based on the goal and user characteristics. General heuristics are used to perform this operation of tying in the user simulation with the domain- and system-specific ontology.

The output of the action planner is an utterance plan including a dialog act type and additional information. This is used in the generator to produce an answer string of the user simulation. The generator follows a simple template-based approach. Our corpus shows that by far the largest part of user turns in commercially deployed spoken dialog systems consist of just a single word. Thus, a very simple baseline generator just outputting single words or short phrases (e.g., number sequences) seems sufficient for reasonably realistic generation. In the future, we intend to extract templates of longer user utterances from the corpus in order to improve on the generator's performance and in order to make its output more variable for testing purposes.

An out-of-the-box text-to-speech system is used to render the generated utterances in spoken German, which is then sent on to the SDS per telephone.

The actual usability evaluation of the SDS is performed in a separate module that can keep track of the incoming utterances and their analysis, as well as the outgoing messages and internal state (e.g., the current user characteristics). The evaluation is based only on objective measures like dialog act types, turn durations, etc. and data derived from these measures, since user judgments as for example in the PARADISE evaluation metric (Walker, Kamm, and Litman 2000) cannot be obtained. The details of this usability evaluation are not the focus of this paper, however.

4. A Human-Machine Dialog Corpus

Development of spoken dialog systems takes time, because the rules and knowledge bases for a new system must be acquired in one of two ways: In a hand-crafted system, which includes virtually all current commercially deployed systems, all rules and knowledge bases must be specified by a human expert. This requires expert knowledge by the designer not only of the underlying dialog platform and architecture, but also about the content domain and interaction structure of the planned dialog system. As an alternative to hand-crafted systems, the strategies in a SDS may be learned automatically from available corpora. Much research has been done in this area recently, especially on dialog strategy optimization by reinforcement learning with (Partially Observable) Markov Decision Processes ((PO)MDPs) (see for example (Lemon and Pietquin 2007) for an overview). This approach works best for learning very specific decisions such as whether or not to ask a confirmation question or how many pieces of information to present to a user (Rieser and Lemon 2007). In addition, such systems must have access to large corpora of interactions with the particular system for training, creating a chicken-and-egg problem. The goal of SpeechEval, however, is to be able to interact with a new SDS in a new domain with little modification. In particular, SpeechEval should be able to evaluate a prototype SDS for which no specialized corpus of human-SDS interactions exists. Therefore, we aim to learn general strategies of user behavior as well as other kinds of knowledge bases for the SpeechEval system from a general dialog corpus.

Since we could not identify an appropriate human-machine dialog corpus in German, we are currently in the process of compiling and annotating the VOICE Awards corpus, which will be a large collection of recordings of dialogs with SDSs from all possible commercially deployed domains. It is based on the "VOICE Awards" competition of German language SDSs.

The annual competition "VOICE Awards"[1] is an evaluation of commercially deployed spoken dialog systems from the German speaking area. Since 2004, the best German spoken dialog applications are entered in this benchmarking evaluation, where they are tested by lay and expert users. We are currently in the process of constructing an annotated corpus of the available audio recordings from this competition, including the years 2005–2008.

The corpus represents a large breadth of dialog systems and constitutes a cut through the current state-of-the-art in commercially deployed German SDSs. Altogether, there are 130 dialog systems in the corpus, with about 1900 dialogs. In each year of the competition, 10 lay users were asked to call the dialog systems to be tested and perform a given task in each of them. The task was pre-determined by the competition organizers according to the developers' system descriptions, and these tasks are usually the same for all 10 lay users. After completing the task, the users filled out satisfaction surveys which comprised the bulk of the evaluation for the award. In addition to these lay callers, two experts interacted with each system and performed more intensive tests, specifically to judge the system's reaction to barge-ins, nonsensical input, etc. These interactions are only in some cases included in the corpus. Table 1 contains a list of some of the domains represented by the dialog systems included in the VOICE Awards corpus.

Audio data for the VOICE Award corpus is available in separate .wav files for each dialog. The transcription of the corpus, using the open source Transcriber tool[2], is about 50% complete. With the transcription, a rough segmentation into turns and dialog act segments is being performed. Since more fine-grained manual timing information is very

[1] http://www.voiceaward.de/

[2] http://trans.sourceforge.net/

public transit schedule information
banking
hotel booking
flight info confirmation
phone provider customer service
movie ticket reservation
package tracking
product purchasing

Table 1: Some domains of SDSs included in the VOICE Awards corpus.

difficult and time-consuming to obtain, it is planned to retrieve word-level timing by running a speech recognizer in forced alignment mode after the transcription is completed.

As a basis of our statistical analyses, the entire corpus is being hand-annotated with several layers of information: (1) Dialog acts, (2) sources of miscommunication, (3) repetitions, and (4) task success. Since the lack of space prohibits a detailed discussion, the annotation schemas are simply listed in table 2. We are using a modified tool from the NITE XML Toolkit (NXT)[3] that has been adapted to our needs to perform these annotations in a single step. The result will be a large corpus of human-SDS-dialogs from many different domains, covering the entire breadth of the current state-of-the-art in commercially deployed German SDSs.

Several other layers of annotation will be added automatically for purposes of saving time, error reduction and consistency. This includes objective information that can be reliably estimated directly from the corpus, such as user reaction time, style and length of user utterances, etc. Some of these automatic annotations are listed in table 3.

dialog acts	hello
	bye
	thank
	sorry
	open_question
	request_info
	alternative_question
	yes_no_question
	explicit_confirm
	implicit_confirm
	instruction
	repeat_please
	request_instruction
	provide_info
	accept
	reject
	noise
	other_da
miscommunication	not_understand
	misunderstand
	state_error
	bad_input
	no_input
	self_correct
	system_command
	other_error
repetition	repeat_prompt
	repeat_answer
task_success	task_completed
	subtask_completed
	system_abort
	user_abort
	escalated
	abort_subtask

Table 2: Hand-annotation schemas of the VOICE Awards corpus.

5. Corpus-Assisted Creation of SDS Resources

As one of the major goals of the SpeechEval systems is easy portability across systems (to be evaluated) and domains, many of the knowledge bases and resources must be learned from corpora. The main corpus for our development is the VOICE Awards corpus described above, which presents a cross-section through many current SDSs. In this section, we describe how this corpus is being used, along with some supplementary sources, to derive the knowledge bases that are part of the SpeechEval architecture (see section 3).

ASR Grammar

In order to improve the coverage of SpeechEval's speech recognition, the recognizer's grammar must be augmented by adding both domain specific terminology as well as terms and phrases that are important in the scenario of spoken dialog systems in general. Different strategies will be used to extract both kinds of vocabulary from the VOICE Awards Corpus as well as other sources.

For the extraction of domain specific terminology, we have categorized the systems in the corpus into domains. A simple chi-square test is used to determine whether a certain word i is significant for a domain j: Given the number of times i occured in j (O_{ij}) and the expected frequency of i in j according to the distribution in the entire corpus (E_{ij}), the chi-square value of the word i for the domain j is computed using the following formula:

$$\chi^2 = \Sigma_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

where the expected frequencies E_{ij} are computed using the following occurrence counts, and formula 2:

	domain j	¬ domain j
word i	a	b
¬ word i	c	d

$$E_{ij} = \frac{(a+c) \times (a+b)}{(a+b+c+d)} \tag{2}$$

Using a stop-word list of the 1000 most frequent terms in German, any word with a chi-square value greater than 3.84 (and whose observed count is higher than the expected one) is likely ($p < 0.05$) to be significant for the domain. Words

[3] http://groups.inf.ed.ac.uk/nxt/

dialog length	time
length of turns	time
# dialog turns	# interactions
	# sds prompts
	# user turns
user reaction time	(by forced alignment)
style of user utterance	single word
	phrase
	full sentence
	SDS-specified / free option

Table 3: Automatic annotations of the VOICE Awards corpus.

which occurred less than 5 times in the corpus were discarded since the test is likely to be inaccurate. This method yielded very good results even when evaluated on a very small subcorpus. Table 4 shows the top 15 positively significant words for the banking domain, as computed on only 58 dialogs from the domain, and a similar amount of out-of-domain dialogs. The only false hits are words that are very suggestive of customer service SDSs in general (e.g., "möchten" / "would like"). These can be excluded by a second stop word list, but they would also be very likely to disappear when a larger amount of data (i.e., the entire VOICE Awards corpus) is used in the computation.

term	**English**	χ^2
Kontostand	account balance	56.6
Kontonummer	account number	54.5
möchten	would like	44.1
Umsätze	transactions	40.7
Konto	account	40.2
Überweisung	wire transfer	32.9
Cent	Cent	29.1
minus	negative	28.1
Ziffer	digit	27.6
Geburtsdatum	birth date	26.0
Hauptmenü	main menu	23.9
Bankleitzahl	routing number	22.9
Servicewunsch	service request	21.8
beträgt	amounts to	21.3
Gutschrift	credit	20.8

Table 4: Significant words in the banking domain.

We plan on extracting SDS-specific terminology (such as "customer id", "main menu", etc.) using the same methodology. All dialogs in the VOICE Awards corpus can be used as the positive subcorpus. For the negative examples, we will use text extracted from web pages representing a similar range of topics and domains as the VOICE Awards corpus. This will ensure that only terminology specific to the medium of spoken dialog systems is marked significant by the chi-square test, and not other frequent content words such as domain-specific terms.

User Characteristics

In order to perform realistic testing of dialog systems, the user simulation's behavior must be relatively varied. We aim to identify suitable user types from the VOICE Awards corpus to model them in our user simulation. Broad distinctions such as expert vs. novice users are known from the literature, but aren't easily observable in the corpus, since by far most dialogs are by lay users. Thus, we instead try to distinguish objectively observable characteristics such as the user reaction time, number of barge-ins, etc. We will perform a clustering on each of these variables in order to obtain a "user properties vector" for each caller in the corpus. The obtained user characteristics then become part of the dialog state vector which determines the following user actions. This will account for the differences in behavior of different user types.

Dialog Act Segmentation and Classification

Machine learning approaches are the standard approaches to the tasks of dialog act segmentation of classification. Good results can be obtained when the number of classes is not too high, although the quality of the ASR output has a large impact on the accuracy, as well. In SpeechEval, we only distinguish 17 mutually exclusive dialog act types (see table 2). Further, the types can be grouped into a flat hierarchy of broad categories such as "question" and "answer". Thus, even in cases where an incoming dialog act has been wrongly classified, SpeechEval's reply may still be appropriate if the misclassified type is of the same super-category.

Our segmentation and classification follows closely the method developed in the AMIDA project (AMIDA 2007). We use the WEKA toolkit to implement separate segmentation and dialog act classification learners. As opposed to this previous work, we use the learned classification modules within an online system. This means that we cannot make use of dynamic features that require the knowledge of future assignments (as is done in the dialog act classifier). Each determined dialog act type is passed on immediately down the pipeline architecture and is acted upon in further modules. However, the reassignment of dialog act labels as done in the work of Germesin (2008) can be used in SpeechEval to retroactively change the dialog history. This may affect both the computation of later dialog act types as well as the confidence scores of SpeechEval's replies.

User Utterance Templates

As noted above, by far most user utterances in our corpus consist of just one word. In an initial study, only 12% of the user turns contained more than one word (number sequences such as ID or telephone numbers were excluded). Most of these longer utterances were false starts or two-word names such as a person's first and last name. Thus, a very simple user simulation baseline will just output the one word which constitutes the answer to the prompt.

For genuine more-word utterances, we are exploring a grammar induction technique in order to extract possible user utterance templates from our corpus. User utterances will be POS-tagged and the possible phrase structures are extracted. In order to find templates, we use our lists of domain-specific words as determined by the chi-square test described above. Domain words can thus be matched onto one another, and general templates with blanks can be extracted this way. The blank spaces are linked to the domain ontology. During generation, the blanks are filled from the ontology if such a template is chosen as a user utterance. With this method, even the rarer longer user utterances can be generated. The advantage is that the system designer does not have to hand-specify a list of possible user utterances in the domain. Instead, general templates are extracted which can be filled with domain vocabulary.

6. Hand-Specified Resources

Even though a goal of SpeechEval is the minimization of hand-crafted resources, certainly not everything can be automatized. In particular, a domain expert must specify the domain ontology which contains the available objects and relations in the domain. The automatically extracted domain vocabulary can be a basis of this ontology, but the relations must be specified by hand.

Further, the set of possible goals which SpeechEval is to pretend to solve must also be pre-specified. This is not surprising. In the VOICE Awards contest, the human judges are also given scenarios to solve for each system. The set of goals to be tested represents the scenario information for the computer evaluator (SpeechEval). During each dialog, one goal is chosen from the set.

7. Conclusion

In this paper we presented the SpeechEval system, a simulation environment that makes possible the quantification of the usability of spoken dialog systems with minimal use of human evaluators and hand-crafted resources. We presented SpeechEval's simple pipelined architecture, with a special focus on the necessary knowledge bases and resources.

In the second part of the paper, we introduced our corpus of German human-machine dialogs, which constitutes the basis of our statistical methods for extracting knowledge bases for spoken dialog systems. We discuss how most of the resources in the SpeechEval architecture, from the ASR grammar to dialog strategy, can be derived from the general dialog corpus or other supplementary corpora. This ensures easy portability of the SpeechEval user simulation across SDSs and domains.

We are currently integrating the system components and carrying out feasibility experiments. The full system will allow speedy evaluation of SDSs during development as well as after updates to deployed systems without the need for large specialized corpora or expensive human evaluators.

8. Acknowledgements

This work is part of the project “SpeechEval: Automatic Evaluation of Interactive Speech-Based Services on the Basis of Learned User Models”, which is being carried out in cooperation with the Quality and Usability Lab at Technical University Berlin. SpeechEval is funded by the Investitionsbank Berlin through the ProFIT framework, grant #10140648. This project is being co-financed by the European Union (European Regional Development Fund).

We would like to thank the three anonymous reviewers for their detailed and helpful comments. All errors remain our own.

References

Ai, H.; Litman, T.; and Litman, D. 2007. Comparing user simulation models for dialog strategy learning. In *Proceedings of NAACL/HLT 2007*, 1–4.

Alexandersson, J., and Heisterkamp, P. 2000. Some notes on the complexity of dialogues. In *Proceedings of the 1st Sigdial Workshop on Discourse and Dialogue*, volume 10, 160–169.

AMIDA. 2007. Deliverable D5.2: Report on multimodal content abstraction. Technical report, DFKI GmbH. chapter 4.

Chung, G. 2004. Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, 63–70.

Georgila, K.; Henderson, J.; and Lemon, O. 2005. Learning user simulations for information state update dialogue systems. In *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech)*.

Germesin, S. 2008. Determining latency for on-line dialog act classification. In *MLMI'08*.

Jung, S.; Lee, C.; Kim, K.; Jeong, M.; and Lee, G. 2009. Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech and Language* 23:479–509.

Lemon, O., and Pietquin, O. 2007. Machine learning for spoken dialogue systems. In *Proceedings of Interspeech*.

López-Cózar, R.; de la Torre, A.; Segura, J.; and Rubio, A. 2003. Assessment of dialog systems by means of a new simulation technique. *Speech Communication* 40:387–407.

Pfalzgraf, A.; Pfleger, N.; Schehl, J.; and Steigner, J. 2008. Odp: Ontology-based dialogue platform. Technical report, SemVox GmbH.

Rieser, V., and Lemon, O. 2007. Learning dialogue strategies for interactive database search. In *Proceedings of Interspeech*.

Schatzmann, J.; Weilhammer, K.; Stuttle, M.; and Young, S. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*.

Schehl, J.; Pfalzgraf, A.; Pfleger, N.; and Steigner, J. 2008. The Babble-Tunes system - Talk to your iPod! In *Proceedings of the 10th international conference on Multimodal interfaces*.

Walker, M.; Kamm, C.; and Litman, D. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering* 6:363–377.

Witten, I., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann, 2nd edition.

Author Index