Natural Language Communication with Social Robots for Assisted Living

Maitreyee Tewari
Department of Computing Science
Umeå University
Umeå, Sweden
maittewa@cs.umu.se

Suna Bensch
Department of Computing Science
Umeå University
Umeå, Sweden
suna@cs.umu.se

***Abstract*—We explore a new dialogue modelling approach for assistive social robots that enables us to formalize flexible conversation flows between a robot and a human. We achieve this by introducing an expectation mechanism to handle, for example, topic change, clarification questions or misunderstandings during a dialogue. The model gave us insight into the dialogue structure and how it is shaped by several linguistic and pragmatic features. This is work in progress and in the future we will explore learning algorithms that mine the features, implement and validate the model with real conversations.**

***Index Terms*—natural language communication, robots, assisted living, dialogue modelling, turn-taking, cooperation, human-robot interaction.**

I. INTRODUCTION

The demographic trend of an aging population is a challenge for the health care system in western countries. Social robots as assistive technology can support care-givers and enable older adults to live longer independently at home and improve quality of life [1]. An essential attribute of assistive social robots is to be natural and efficient for us to use. An intuitive interaction may involve the robot to interpret and react to human behaviour including gesturing, displaying emotions, and using natural language to conduct a dialogue (we focus only on natural language aspects of interaction). A social robot used in the context of elder care has to adapt to the varying and unpredictable nature of dialogues, such as sudden topic changes, misunderstandings, incomplete or inaccurate information (non-understanding), interruptions, humour and opposition. To describe this variability we introduce a new model that formalizes flexible dialogue flows between a robot and a human and provides *insight* into the dialogue structure.

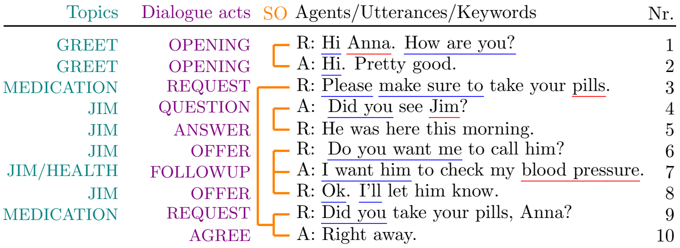
The model *co-operating distributed grammar systems with expectations* ($\mathrm{CDGS}_{\mathrm{exp}}$ for short) is based on co-operating distributed grammar systems (CDGS) [2]. Such systems model cooperation among several agents that have a common goal. We consider a dialogue between a robot and a human as cooperation between two agents who have the common goal of conducting a successful dialogue. In the latter we refer to dialogue participants as agents. Expectations are anticipations of certain information that agents have when conducting a dialogue. For example, an agent A can expect that another agent B confirms agent A's request or answers agent A's question. We formalize expectations as internal control mechanism that is bound to a given time frame. The time frame can be a measure of the number of turn takes during a dialogue or discrete time unit steps. The internal control mechanism allows for flexible dialogue flows as it gives agent the possibility to not meet expectations immediately but, for example, change the current topic of conversation. $\mathrm{CDGS}_{\mathrm{exp}}$ allow us to control the dialogue flow according to the agents' expectations, to describe the agents' perspective during a dialogue, and to model the overall dialogue structure and its formation. Several linguistic and pragmatic features that influence the dialogue structure are also illustrated.

II. BACKGROUND

Dialogue management approaches are generally divided into finite-state based, and data-driven approaches [3]. Data-driven approaches use machine learning methods to learn from data on how to conduct a dialogue and deliver often fast and useful results [4]. The learning outcomes of data-driven approaches are often non-transparent which we find, in the context of developing dialogue management techniques for assistive social robots, not suited (for reasons outlined in the introduction). Finite-state based approaches manually define how to conduct a dialogue and thus provide valuable insight into the dialogue structure, but manual definition of dialogue rules is time and labor costly.

We are interested in developing a hybrid dialogue management approach that learns from data (certain aspects of) the dialogue structure and give us full control over adding, deleting or altering dialogue rules. In [5] author proposes a hybrid dialogue model based on probabilistic rules, where these rules act as templates to generate a graphical model for dialogue management. As a first step, we focus on developing a suitable formal model for dialogues management based on co-operating distributed grammar systems which are finite-state devices. Later in the future we will learn the pragmatic and linguistic features with data driven methods. In [6] authors propose *conversational grammar systems*, which mimics natural language to define a formal model for dialogues.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 721619 for the SOCRATES project.



Topics	Dialogue acts	SO	Agents/Utterances/Keywords	Nr.
GREET	OPENING		R: Hi Anna. How are you?	1
GREET	OPENING		A: Hi. Pretty good.	2
MEDICATION	REQUEST		R: Please make sure to take your pills.	3
JIM	QUESTION		A: Did you see Jim?	4
JIM	ANSWER		R: He was here this morning.	5
JIM	OFFER		R: Do you want me to call him?	6
JIM/HEALTH	FOLLOWUP		A: I want him to check my blood pressure.	7
JIM	OFFER		R: Ok. I'll let him know.	8
MEDICATION	REQUEST		R: Did you take your pills, Anna?	9
	AGREE		A: Right away.	10

Fig. 1. A fictional sample dialogue between a robot (R) and an older adult named *Anna* (A). The dialogue is analyzed taking into account several linguistic and pragmatics features, namely *topics*, *dialogue acts*, *sequence organization* (SO) and *keywords* (which are underlined).

In [7] turn-taking behavior in dialogues is modelled with *CDGS with memories*. We extend CDGSs with an expectation mechanism and model linguistic and pragmatic features to derive a dialogue structure.

III. METHODOLOGY

A. Linguistic and pragmatic features for dialogue analysis

We consider dialogues as sequences of utterances, consisting of one or more sentences, aligned one after the other by participants through turn-takes. Consider the fictional dialogue in Figure 1 between a robot (R) and an older adult named *Anna* (A) in a health care facility. The dialogue is displayed in the fourth column "Agents/Utterances". We refer to the individual utterances with numbers which are displayed in the fifth column "Nr.", where an utterance can consist of one or more sentences. The dialogue starts with the two agents greeting each other (Utterances 1-2). Then the robot reminds Anna politely to take her pills (Utterance 3). Anna instead of answering the request (Utterance 3), changes the topic by asking whether the robot has seen Jim (Utterance 4). The robot answers Annas' question and offers to call Jim (Utterances 5-6). Anna then states that she wants Jim to check her blood pressure (Utterance 7) which is indirectly also an acceptance of the robot's offer to call Jim. The robot confirms that it will let Jim know that Anna wants to see him (Utterance 8). Then the robot reminds Anna again about her medicine (Utterance 9) which Anna promises to take right away (Utterance 10).

We analyze a dialogue considering the following linguistic and pragmatic features of utterances: topics, dialogue acts, sequence organization and keywords. We explain all four features briefly in the following.

The first column in Figure 1 shows some topics of the utterances. The second column shows the so-called dialogue acts [8] associated with each utterance. An utterance is a dialogue act if it has a communicative function, which specifies an activity performed in the dialogue such as asking a question, requesting information, accepting or rejecting a request or making a declaration. For example, in Figure 1 the dialogue act OFFER that can be associated to the utterance *"Do you want me to call him?"* has the communicative function of offering (to do an action). The third column in Figure 1 illustrates a possible sequence organization of the utterances in the dialogue. Sequence organization (SO for short) is empirically studied in *conversation analysis* [9]. Sequence organization describes how sequences of utterances can be ordered. In Figure 1 utterances forming a sequence with each other are connected by the displayed orange lines. For example, if two utterances occur consecutively (e.g. question-answer, greeting-greeting) then they can be described as so-called *adjacency pair*. In Figure 1, Utterances 1 and 2 form an adjacency pair. Utterances do not have to be necessarily adjacent to each other, they can occur apart from each other in a dialogue and are then ordered as *First-Pair-Part* (FPP) and *Second-Pair-Part* (SPP). Furthermore, sequences of utterances can be categorized into three types of so-called *expansions*, namely base, insert and post [9]. In Figure 1, Utterance 3 (which is a particular kind of FPP, namely FPP_{base}) is connected to the adjacency pair made up of Utterances 9-10 (which is the corresponding SPP, namely SPP_{base}). The topic change (i.e. *Did you see Jim?*) by Anna expands the so-called base sequence (i.e. FPP_{base} and SPP_{base}) and introduces $\text{FPP}_{\text{insert}}$, which is followed by the robots response generating $\text{SPP}_{\text{insert}}$ (i.e. Utterances 4 and 5, respectively). Another feature that can influence how a dialogue is structured are frequently used words or phrases (i.e. keywords). In Figure 1 in the fourth column, keywords are underlined in blue or red (for domain specific words or phrases). Such keywords can facilitate, for example, dialog act or topic association and thus influence the structuring of the dialogue.

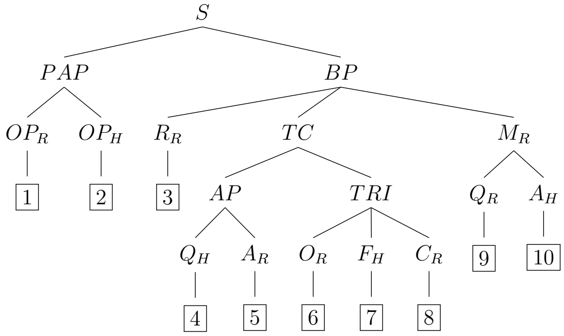


Fig. 2. The tree structure of the dialogue displayed in Figure 1. The leaf nodes are labelled with the numbers of the utterances in Figure 1.

B. Inferring the tree structure of the dialogue

All the features elaborated so far (e.g. topics, dialogue acts, keywords, sequence organization) are organized into a tree structure which is illustrated in Figure 2. A tree structure serves the following two purposes:

1) To describe the overall dialogue structure based on the linguistic and pragmatic features, and
2) To extract rules for our model CDGS_{exp}.

The tree in Figure 2 illustrates that our example dialogue consists of two larger parts, namely an introduction into the dialogue represented by the subtree rooted at the node labelled by PAP (e.g. greeting and asking about well-being) and a main part represented by the subtree rooted at the node labelled by BP. The topic change is represented by the subtree rooted at the node labelled by TC. The leaves of the tree are labelled with the numbers of the individual utterances that can be found in Figure 1. Reading the labels off the leaves, from left to right, restores the chronological order of the uttered sentences in the dialogue. The nodes that are parents of the leaf nodes, (e.g. $OP_R, OP_H, R_R, Q_H, A_R$) are labels for the dialogue acts and we can restore the order in which they were uttered too. Note that the subtree with root label TC is a subtree that can only be formed by taking into account the information about the topic change (i.e. sequence organization alone would not be enough). How a tree as in Figure 2 can be generated by our CDGS_{exp} is explained later.

C. Formal background

In this section we provide the necessary definitions of cooperating distributed grammar systems (CDGSs). A CDGS consist of several so-called *components* that *work by taking turns* according to some *cooperation protocol*. The cooperation protocol defines when components are allowed to start and stop working. The components in a CDGS can be interpreted as agents working together with a *common aim* (e.g. to solve a problem).

Definition 1: A CDGS of degree n, with $n \geq 1$, is an $(n+3)$-tuple $G = (N, T, C_1, C_2, \ldots, C_n, S)$, where, N is a set of variables (called non-terminal symbols), T is a set of constants (called terminal symbols), S is the start symbol, for $1 \leq i \leq n$, C_i is a set of rules of the form $A \rightarrow \alpha$, where $A \in N$ and α is a string consisting of variables and/or constants (i.e. $N \cup T$). A rule $A \rightarrow \alpha$ means that a variable A can be replaced with the string α. The set of rules $C_1, C_2, \ldots, C_n$ are called *components*.

Example 1: Let $\hat{G} = (\{S, A, B\}, \{a, b, c, d\}, C_1, C_2, S)$ be a CDGS grammar, where

$$C_1 = \{S \rightarrow aA, B \rightarrow aA, A \rightarrow aA, A \rightarrow a\},$$

$$C_2 = \{S \rightarrow bB, A \rightarrow bB, B \rightarrow bB, B \rightarrow b\}.$$

Definition 2: Let $G = (N, T, C_1, C_2, \ldots, C_n, S)$ be a CDGS. For two strings x, y in $(N \cup T)$ and $1 \leq i \leq n$, we write $x \Longrightarrow_i y$ and say that y is derived in one derivation step from x by component C_i, if and only if $x = \gamma_1 A \gamma_2$ and $y = \gamma_1 \alpha \gamma_2$ for some $\gamma_1, \gamma_2 \in (N \cup T)$ and there exists a rule in C_i of the form $A \rightarrow \alpha$. A *derivation* (i.e. successive derivation steps) starts with the string S (i.e. the start symbol of G) and ends when a string w is obtained that consists only of terminal symbols.

The cooperation protocol for a CDGS can state that a component can make exactly k derivation steps, $\leq k$ steps, $\geq k$ steps, arbitrary many steps ($*$ cooperation protocol) or take the maximal number of derivation steps possible (t cooperation protocol).

Example 2: Let $\hat{G}$ have the cooperation protocol $= 2$, that is, each component must make exactly two derivation steps before the other component starts to work. The derivation starts with the start symbol S. Both components C_1 and C_2 can rewrite the start symbol S (by applying the rules $S \rightarrow aA$ or $S \rightarrow bB$, respectively). Let us assume that C_1 starts to work. The component C_1 has to make two derivation steps, $S \Longrightarrow_1 aA \Longrightarrow_1 aaA$, that is, first rewriting S by applying the rule $S \rightarrow aA$ and then rewriting A (in the string aA) by applying the rule $A \rightarrow aA$. Now component C_2 has to start rewriting and make two derivation steps. Let us assume the derivation $aaA \Longrightarrow_2 aabB \Longrightarrow_2 aabb$. That is, C_2 applied the rule $A \rightarrow bB$ to the string aaA generating the string $aabB$ and then applied the rule $B \rightarrow b$ to the string $aabB$ generating the string $aabb$. The string $aabb$ is a *terminal string* and consists only of terminal symbols and cannot be rewritten further.

This example illustrated how components generate strings by taking turns after two derivation steps according to the given cooperation protocol. Note that the components can generate many different terminal strings (e.g. $aabb$, bb, $aabbaabb$).

D. CDGS modeling expectations

In this section we provide the definitions of our new model CDGS_{exp} and apply it to the dialogue example in Figure 1 and show how the tree in Figure 2 is generated. We assume that a CDGS_{exp} works in $*$ cooperation protocol with the addition that an agent A starts working if the other agent B did not meet the expectation of agent A within a given time frame. An agent A stops working whenever it is ready to "hand the floor" to agent B. In a CDGS_{exp} a non-terminal symbol A on the right hand side of a rule may be extended with $\leq k$, where k is a positive integer, that is, $A[\leq k]$. The $\leq k$ in $A[\leq k]$ represents the time frame in which the other component is expected to rewrite the non-terminal A. The time frame can be a measure of the number of turn takes during a dialogue or of derivation steps. In the following example, we count the number of derivation steps each agent makes[1]. For example, if an agent C_1 applies a rule of the form $B \rightarrow aA[\leq 5]$, it represents that agent C_1 expects the other agent C_2 to rewrite symbol A within the next 5 derivation steps C_2 makes. If the other component does not rewrite the non-terminal that is expected to be rewritten within the given time frame, then the component that has the expectation starts working and applies a rule with the same expectation. That is, if, for example C_1 applied the rule $B \rightarrow aA[\leq 5]$ and the component C_2 does not rewrite A within five steps, then component C_1 applies a rule $A \rightarrow aA[\leq 2]$ (i.e. now C_1 expects C_2 to rewrite the symbol A within its next two derivation steps). Let $\gamma_1 A \gamma_2$ be a string for some $\gamma_1, \gamma_2 \in (N \cup T)$ and let $r : A \rightarrow \alpha[\leq k]$ be a rule r in a component C_i. Then C_i derives y by applying r as follows: $\gamma_1 A \gamma_2 \Longrightarrow_i \gamma_1 \alpha \gamma_2 = y$. That is, $[\leq k]$ is not introduced into a string but only appears in C_i.

In our scenario where we consider assistive robots with conversational capabilities, this serves the purpose to give the older adult the freedom to react flexibly, and at the same time, ensure that the robot picks up a topic again if it has not been

---

[1] Note that we can just as easily count the number of turns each agent makes by defining a turn of an agent A as an application of a rule of the form $A \rightarrow a$, where $a \in T$ for a given CDGS_{exp} and component A.

answered by the older adult (see Utterance 3 and Utterance 9 in Figure 1).

The following example is simplified but should give the idea of how the tree in Figure 2 is generated in cooperation between an agent C_1 (representing the robot) and an agent C_2 (representing the human). We assume that the CDGS$_{\text{exp}}$ works in a leftmost derivation fashion, that is, it always rewrites the leftmost occurring symbol in a string. We can associate to each derivation a derivation tree. Roughly speaking, every applied rule during the derivation is displayed in the derivation tree. For example, if the rule $S \to PAP\ BP$ is applied then a subtree t can be associated with the root labelled S and the two children labelled PAP and BP.

Example 3: Let $\hat{\hat{G}} = (N, T, C_1, C_2, S)$ be a CDGS$_{\text{exp}}$, where $N = \{S, PAP, BP, OP_R, OP_H, R_R, TC, M_R, AP, TRI, Q_R, A_H, Q_H, A_R, O_R, F_H, C_R\}$ (that is, all labels of the inner nodes in the tree in Figure 2), $T = \{\boxed{1}, \boxed{2}, \ldots, \boxed{10}\}$ (that is, all utterances given in Figure 1) and C_1 and C_2 contain the rules shown in Figure 3 (we number all rules for easier reference):

C_1	C_2
$\{r_1 : S \to PAP\ BP,$	$\{r_1 : OP_H \to \boxed{2},$
$r_2 : PAP \to OP_R\ OP_H[\leq 1],$	$r_2 : TC \to AP\ TRI,$
$r_3 : OP_R \to \boxed{1},$	$r_3 : AP \to Q_H\ A_R[\leq 1],$
$r_4 : BP \to R_R\ TC\ M_R[\leq 5],$	$r_4 : Q_H \to \boxed{4},$
$r_5 : R_R \to \boxed{3},$	$r_5 : F_H \to \boxed{7},$
$r_6 : A_R \to \boxed{5},$	$r_6 : A_H \to \boxed{10}\}$
$r_7 : TRI \to O_R\ F_H[\leq 2]C_R,$	
$r_8 : O_R \to \boxed{6},$	
$r_9 : C_R \to \boxed{8},$	
$r_{10} : M_R \to Q_R A_H[\leq 2],$	
$r_{11} : Q_R \to \boxed{9}\}$	

Fig. 3. The components C_1 and C_2 for the CDGS$_{exp}$ $\hat{\hat{G}}$ in Example 3.

The robot initiates the dialogue which is represented by C_1 applying the rules r_1, r_2, r_3 in C_1, that is, $S \Longrightarrow_1 PAP\ BP \Longrightarrow_1 OP_R\ OP_H\ BP \Longrightarrow_1 \boxed{1}\ OP_H\ BP$. The symbol $\boxed{1}$ represents the Utterance 1 in Figure 1 by the robot. The component C_1 expects C_2 to rewrite the symbol OP_H within one derivation step. The component C_2 rewrites OP_H by applying rule r_1 in C_2, i.e. $\boxed{1}\ OP_H\ BP \Longrightarrow_2 \boxed{1}\ \boxed{2}\ BP$. The string $\boxed{1}\ \boxed{2}\ BP$ represents that two utterances have been made and that the symbol BP is to be rewritten by C_1. The component C_1 applies the rules r_4, r_5 generating the string $\boxed{1}\ \boxed{2}\ \boxed{3}\ TC\ M_R$. The variable TC allows C_2 to change the topic. C_2 applies the rules r_2, r_3, r_4 in C_2 to generate the string $\boxed{1}\ \boxed{2}\ \boxed{3}\ \boxed{4} A_R\ TRI\ M_R$. The derivation is continued in this fashion until we obtain the terminal string $\boxed{1}\ \boxed{2}\ \boxed{3}\ \boxed{4}\ \boxed{5}\ \boxed{6}\ \boxed{7}\ \boxed{8}\ \boxed{9}\ \boxed{10}$, that represents the dialogue in Figure 1. Note that the tree in Figure 2 is the derivation tree of the derivation outlined above. In our model, expectations are not restricted to only expecting certain dialogue acts (as in the above example) but can be larger structures (e.g. topic change) too.

IV. CONCLUSIONS AND FUTURE WORK

We present a new approach for modelling dialogues for assistive social robots that allow flexible conversation flows. We achieve this by introducing an expectation mechanism that ensures that dialogue goals are met, but at the same time allows dialogues to divert (e.g. topic change, misunderstanding, clarification questions). Our new approach has the following additional advantages compared to other finite state approaches or data-driven approaches for dialogue modelling:

- We describe dialogue as a cooperation among agents instead of only capturing the machine's perspective.
- We gain insight into the structure of dialogues and in its formation instead of trusting black boxes.
- Our approach is extendable to several agents and can serve as models for human robot communication in which several robots and humans are communicating.

This paper reports work in progress and in the future we want to develop algorithms that learn how to map sets of features such as topics, dialogue acts, keywords, sequence organization into dialogue structures such as the one displayed in Figure 2. Once this is achieved, a CDGS$_{\text{exp}}$ with expectations can be generated. We are interested in further investigating how our model can handle dialogue phenomena such as misunderstandings, non-understandings or opposition. Another of our tasks is an implementation of our formal model to test its validity and limitations.

REFERENCES

[1] R. Li, B. Lu, and K. D. McDonald-Maier, "Cognitive assisted living ambient system: a survey," *Digital Communications and Networks*, vol. 1, no. 4, pp. 229 – 252, 2015.

[2] E. Csuhaj-Varju, J. Kelemen, G. Paun, and J. Dassow, Eds., *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*, 1st ed. Newark, NJ, USA: Gordon and Breach Science Publishers, Inc., 1994.

[3] C. Lee, S. Jung, K. Kim, D. Lee, and G. Lee, "Recent Approaches to Dialog Management for Spoken Dialog Systems," *Journal of Computing Science and Engineering*, vol. 4, pp. 1–22, 2010. [Online]. Available: http://www.dbpia.co.kr/Article/NODE01431934

[4] H. Cuayáhuitl, *SimpleDS: A Simple Deep Reinforcement Learning Dialogue System*. Singapore: Springer Singapore, 2017, pp. 109–118.

[5] P. Lison, "A hybrid approach to dialogue management based on probabilistic rules," *Computer Speech & Language*, vol. 34, no. 1, pp. 232–255, 2015.

[6] G. Bel-Enguix and M. Jiménez-López, "Modelling dialogue as interaction," *International Journal of Speech Technology*, vol. 11, no. 3, pp. 209–221, 2008.

[7] S. Aydin, H. Jürgensen, and L. E. Robbins, "Dialogues as co-operating grammars," *Journal of Automata, Languages and Combinatorics*, vol. 6, no. 4, pp. 395–410, 2001.

[8] H. Bunt, "Dimensions in dialogue act annotation," *Proceedings of Language Resources and Evaluation Conference*, vol. 6, pp. 919–924, 2006.

[9] J. Sidnell and T. Stivers, *The handbook of conversation analysis*. John Wiley & Sons, 2012, vol. 121.