A Dialogue System for Indoor Wayfinding Using Text-Based Natural Language

Heriberto Cuayáhuitl[1], Nina Dethlefs[2], Kai-Florian Richter[1], Thora Tenbrink[12], and John Bateman[12]

[1] Transregional Collaborative Research Center, SFB/TR 8, University of Bremen
[2] FB10 Faculty of Linguistics and Literary Sciences, University of Bremen
heriberto@uni-bremen.de

Abstract. We present a dialogue system that automatically generates indoor route instructions in German when asked about locations, using text-based natural language input and output. The challenging task in this system is to provide the user with a compact set of accurate and comprehensible instructions. We describe our approach based on high-level instructions. The system is described with four main modules: natural language understanding, dialogue management, route instruction generation and natural language generation. We report an evaluation with users unfamiliar with the system — using the PARADISE evaluation framework — in a real environment and naturalistic setting. We present results with high user satisfaction, and discuss future directions for enhancing this kind of system with more sophisticated and intuitive interaction.

1 Introduction

Wayfinding in (partially) known environments poses a considerable challenge for humans. This fact is not only confirmed by a substantial body of research [1, 2] but also by the ubiquity and high demand for incremental navigation assistance systems, as well as web-based services providing in-advance information about routes. However, most information provided by such systems is tailored for large-scale navigation using cars or public transport [3]. Indoor wayfinding assistance is not a trivial issue and has not been addressed widely so far. Related work includes the following. Kray et al. [4] present an interactive display system mounted on walls providing visual navigation support to building users. Callaway [5] describes indoor navigation help while navigating rather than in-advance directions as explored here. A modelling software proposed by Münzer and Stahl [6] generates dynamic visual route information. Hochmair [7] reports a desktop usability study comparing various modes of indoor navigation aids. Becker et al. [8] and Ohlbach and Stoffel [9] present models for representing the complex spatial configurations adequately for navigation and route assistance. Kruijff et al. [10] present and discuss a human-robot interaction scenario set within an office environment. Automatic systems generating natural language-based route descriptions in-advance have therefore received little attention to date.

In the following we present a first substantial step in this direction: a dialogue system that automatically generates indoor route instructions in German when asked about locations, using text-based natural language input and output. The challenging task in this system is to provide the user with a compact set of accurate and comprehensible instructions suitable for navigating in a complex indoor setting. Our test environment is a campus building which, due to a range of asymmetries and unconventional architectural features, poses a range of navigational challenges.

2 System architecture

This dialogue system aims to provide users with route descriptions in German for navigating in a particular building of our university that is generally recognised as presenting significant navigational challenges to both new and infrequent visitors. A pipeline architecture of this system is shown in the high-level diagram of Figure 1. First, the user interacts with a Graphical User Interface (GUI) by asking questions about route directions using text-based natural language. Second, the language understanding module applies OpenCCG parsing [11] and keyword spotting — the latter is used in case of unparsed inputs — to the user utterance in order to extract a user dialogue act. Third, the dialogue management module specifies the system's behaviour by mapping knowledge-compact dialogue states (extracted from the knowledge base) to machine dialogue acts such as 'request' , 'clarify' or 'present_info'. Fourth, the language generation module provides high-level route instruction through the use of pCRU that generates logical forms that are then given to the KPML language generator [12], which in turn outputs text to be shown in the GUI (see Figure 2). Finally, the knowledge base maintains the history of the interaction. These modules were integrated under the DAISIE framework, which provides support for building situated dialogue systems [13]. These modules are described in more detail in the rest of this section.

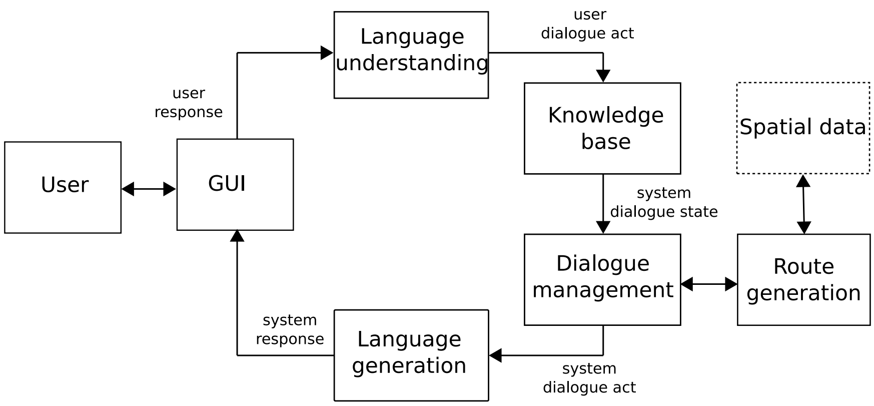

Fig. 1. *A pipeline architecture of our dialogue system for indoor wayfinding.*

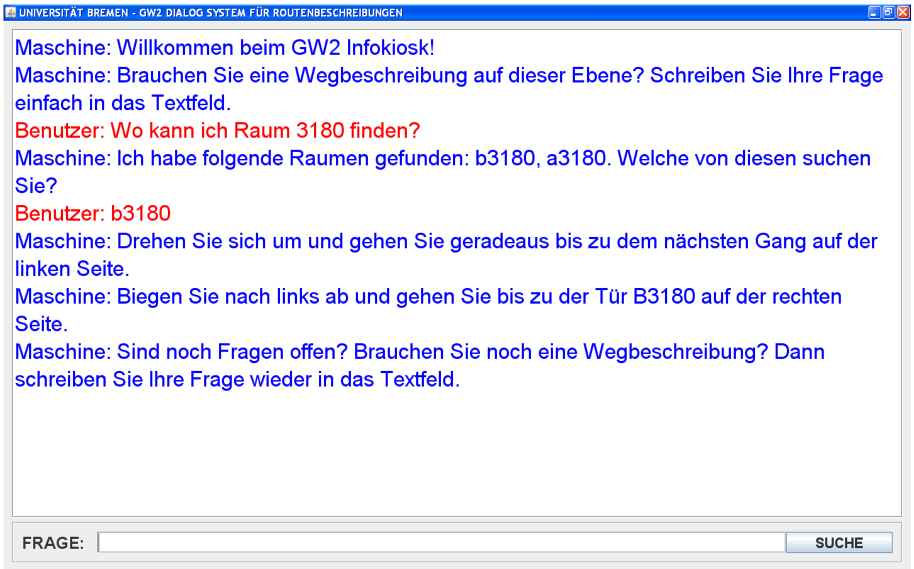

Fig. 2. *A screenshot of our text-based dialogue system for indoor wayfinding. A translation to English of this dialogue is provided in Table 2.*

2.1 Natural language understanding

For parsing the textual user inputs we used the OpenCCG parser [11]. We use a grammar for German and represent the user input in a structure called 'Frame Object Structure' [14], using semantic types derived from the Generalized Upper Model [15]. A sample structure for the sentence 'Wie komme ich zu Raum a3440?' (How do I get to room a3440?) is represented as

```
@g1:gs-NonAffectingDirectedMotion(gum-Coming ^
      <uio-hasSurfaceFunction>uio-Wh-Question ^
      <gum-actor>(p1:gum-ConsciousBeing ^ Pron) ^
      <gum-processInConfiguration>(g2:gum-Process ^ gum-Coming) ^
      <gs-motionDirection>(x1:gs-GeneralizedLocation ^
            <gs-hasSpatialModality>(z1:gs-GeneralDirectionalNearing ^ zu) ^
            <gs-relatum>(r1:slm-Room ^ Raum ^
                        <num>sg ^
                        <Modifier>(a1:slm-Office ^ a3440))))
```

In addition, we used a keyword spotter to identify locations in case of sentences without parse in the CCG grammars. The task of the keyword spotter is to identify names of locations or names of people and to treat the remaining words as fillers. The output of this module is a user dialogue act represented by a used dialogue act type ('ask', 'provide', 'confirm' , 'silence') and slot-value pairs. The dialogue act for the sample above can be described as 'ask(destination=room a3440)'. We used the same format for describing system dialogue acts.

2.2 Dialogue management

Our dialogue manager is based on the Markov Decision Process (MDP) model, but we use a deterministic mechanism for action-selection. The MDP model is used to optimize stochastic sequential decision making problems and is defined as a 4-tuple $< S, A, T, R >$, where S is a finite set of states, A is a finite set of actions, T is a state transition function, and R is a reward function. The solution to an MDP is to find a policy $\pi(s)$ that maps states s to actions a. Because we use deterministic action-selection, we can omit the reward function. This form of control is typically used as baseline for learnt dialogue strategies [16, 17].

We applied this model to our system as follows: (1) the space of dialogue states is represented with a vector of state variables as shown in Table 1, (2) the action space is represented with dialogue act types shown in Equation 1, (3) the state transitions are modelled by observing dialogue states from the knowledge base, and (4) the deterministic dialogue policy is defined in Equation 1. A sample human-machine dialogue illustrating this form of interaction is shown in Table 2. This dialogue is described with wordings in German and English and corresponds to the dialogue shown in Figure 2.

Table 1. *Representation of the dialogue state. Each state variable has either three or four possible values, resulting in* $3^5 \times 4^2 = 3888$ *states.*

Variable	ID	Domain Value
Salutation	X_1	0=null, 1=greeting, 2=closing
Origin	X_2	0=unknown, 1=requested, 2=known
Destination	X_3	0=unknown, 1=requested, 2=known
NumTuples	X_4	0=null, 1=one, 2=more-than-one
Instructions	X_5	0= unknown, 1=known, 2=provided
UserUtterance	X_6	0= unknown, 1=parsed, 2=unparsed, 3=spotted
MoreInstructions	X_7	0=null, 1=empty, 2=yes, 3=no

$$\pi(s) = \begin{cases} \text{opening} & \text{if } s \in \{0000000\} \\ \text{request} & \text{if } s \in \{1000000, 1000012\} \\ \text{other\_request} & \text{if } s \in \{1220210, 1220220\} \\ \text{query\_route} & \text{if } s \in \{1220210, 1220220, 1110030, 1210030, \\ & \quad 1220030, 1211030, 1221030\} \\ \text{present\_info} & \text{if } s \in \{1221110, 1221130, \} \\ \text{clarify} & \text{if } s \in \{1112100, 1112030, 1212030, 1222211, 1222231\} \\ \text{apologize} & \text{if } s \in \{1110020, 1210020, 1220220, 1210210\} \\ \text{confirm} & \text{if } s \in \{1112010, 1112030\} \\ \text{closing} & \text{if } s \in \{1*****3\} \\ \text{wait} & \quad \text{otherwise} \end{cases} \tag{1}$$

Table 2. *Sample dialogue in the wayfinding domain, where the dialogue state is formed by the state variables shown in Table 1, and a set of actions (shown in Equation 1) is available per state. The action-selection policy is defined by Equation 1.*

Agent	Dialogue State	Action	Sample Utterance
SYS	0000000	opening	Willkommen beim <Hidden for Review> Infokiosk! [Welcome to the <Hidden for Review> info kiosk!]
	1000000	request	Brauchen Sie eine Wegbeschreibung auf dieser Ebene? Schreiben Sie Ihre Frage einfach in das Textfeld. [Do you need a route description on this level? Just write your question into the text field.]
USR			*Wo kann ich Raum 3180 finden?* [Where can I find room number 3180?]
SYS	1212030	clarify	Ich habe folgende Raeume gefunden: b3180, a3180. Welche von diesen suchen Sie? [I have found the following rooms: b3180, a3180. Which of these are you looking for?]
USR			*b3180*
SYS	1220010	query_route	<query the route generation module>
SYS	1221110	present_info	Drehen Sie sich um und gehen Sie geradeaus bis zu dem naechsten Gang auf der linken Seite. [Please turn round and go straight on until the next hallway on the left hand side.] Drehen Sie sich nach links und gehen Sie bis zu der Tuer B3180 auf der rechten Seite. [Turn left and go until the door B3180 on the right hand side.]
SYS	1220210	other_request	Sind noch Fragen offen? Brauchen Sie noch eine Wegbeschreibung? Dann schreiben Sie Ihre Frage wieder in das Textfeld. [Do you have further questions? Do you need another route description? In that case please write your question again into the text field.]

2.3 Route instruction generation

Our system employs a computational process called Guard (Generation of Unambiguous, Adapted Route Directions) [18] for generating context-specific route instructions. Context-specific route directions account for environmental characteristics and a route's properties. They adapt communication to the action to be taken in the current surrounding environment. Such instructions are termed "context-specific" because of the explicit adaptation to the structure and function in wayfinding [19]. Guard unambiguously describes a specific route to a destination, with instructions adapted to environmental characteristics. Selection of the route is not part of Guard itself. Guard originally has been developed for providing route instructions in outdoor environments. Figure 3 provides an overview of the generation process.

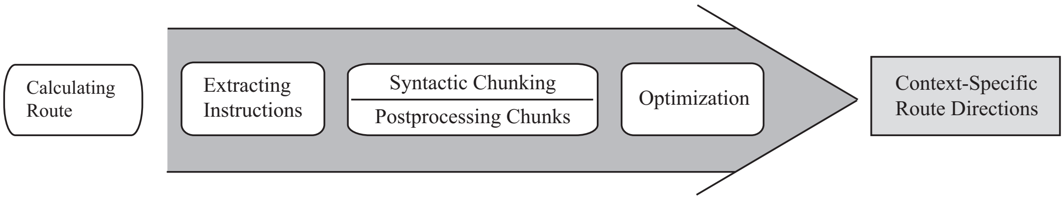

Fig. 3. Overview of GUARD, the generation process for context-specific route directions.

GUARD works on a network representation of paths in an environment. This graph is annotated with information on landmarks, for example, their location and shape. GUARD accounts for different types of landmarks in generating instructions whose role in the route instructions depends on their location relative to the route [20, 21]. Landmarks are associated with decision points based on a heuristic that accounts for distance and potential obstruction. When generating instructions, each associated landmark is tested for whether it can be used as a reference object in the instruction, which depends on its functional role in the given spatial configuration [21].

The generation of context-specific route instructions is a three-step process. First, for every decision point of the route all instructions that unambiguously describe the route segment to be taken are determined. This results in a set of possible instructions for each decision point. Next, GUARD performs spatial chunking. Spatial chunking refers to combining instructions for several consecutive decision points into a single instruction, for example, "turn left at the third intersection" instead of "straight, straight, left." GUARD is flexible with respect to the principles used in chunking (e.g., [22, 3]). Finally, in the third step, the actual context-specific route directions are generated. Here, from all possible instructions, those that best describe the route are selected. As this is realized as an optimization process, "best" depends on the chosen optimization criterion. Just as with the chunking principles, GUARD is flexible with respect to the criterion used. As a default, it aims for instructions that contain the least number of chunks, i.e., that require the least number of individual instructions[18]. Optimization results in a sequence of chunks that cover the complete route from origin to destination. Due to the aggregation of instructions performed in chunking, instructions for some decision points will be represented implicitly, thus, reducing the amount of communicated information.

In summary, the approach to context-specific route directions finds the best instruction sequence according to the optimization criterion, but for a previously given route. Recently, there has been work on using GUARD's principles in a path search algorithm finding the routes that are also the easiest to describe [23].

2.4 Natural language generation

Generation of high-level instructions. Our approach for generating high-level route instructions is described in Algorithm 1. Briefly, it operates with the following steps: (a) it receives the output of the route instruction generator; (b) segments the received low-level instructions based on major changes of direction such as left or right; (c) obtains a landmark and direction for the current segment; (d) generates a turning instruction (cf. line 10); (e) generates a go instruction until the current landmark (cf. line 11); (f) unifies the previous two instructions; and (g) generates the language for the unified instruction (cf. line 13). Whilst steps d and e are processed with the pCRUs described in the next subsection, step g is processed with the KPML language generation system [12]. An example of this process using 'corridors' as non-terminal landmarks is illustrated in Figure 4.

Algorithm 1 Generator of high-level textual route instructions

```
 1: function GeneratorOfHighLevelInstructions(lowLevelInstructions)
 2:     segments ← segment low-level instructions based on major changes of
                       directions such as left and right.
 3:     for each segment do
 4:         if non-terminal segment then
 5:             landmark ← destination landmark for the current segment
 6:         else
 7:             landmark ← target destination
 8:         end if
 9:         direction ← direction of the current landmark (e.g. left, right, in front)
10:         spl1 ← obtain a turning direction (e.g. turn around, turn left, turn right)
11:         spl2 ← obtain a go direction to the landmark with corresponding direction
12:         instruction ← aggregation of spl1 and spl2
13:         Generate the textual description corresponding to the current instruction
14:     end for
15: end function
```

Generation of routes with pCRU. For the generation of route descriptions, we distinguish different route-associated actions that need to be performed in different segments of a route, for example, turning actions or following actions. While these could be verbalised by a template-based approach, we instead use full NLG and aim to make our descriptions more natural by allowing appropriate variation in the realisation of route segments, so as to reflect the same tendencies found in human descriptions. We achieve this by using the pCRU framework described in the rest of this section.

Probabilistic context-free representational underspecification (pCRU) [24] is an approach to resolving the nondeterminacy that typically arises in generation between a semantic representation and its possible linguistic surface forms. This relationship is almost always one-to-many as can be illustrated by the following example. Consider the following SPL [25], which serves as an input to the KPML generation system [12].

```
(v0 / |space#NonAffectingOrientationChange|
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ROUTE>
<Direction>
<instruction point="1" id="307" relation="turn straight">
</instruction>
</Direction>

<Direction>
<instruction point="2" id="308" relation="straight">
</instruction>
<instruction point="3" id="310" relation="half right">
</instruction>
</Direction>

<Direction>
<instruction point="4" id="314" relation="half left">
</instruction>
</Direction>

<Direction>
<instruction point="5" id="192" relation="left">
<landmark id="147" type="ns:exitsign" relation="lm&gt;"/>
</instruction>
</Direction>

<Direction>
<instruction point="6" id="194" relation="straight">
</instruction>
<instruction point="7" id="196" relation="straight">
</instruction>
<instruction point="8" id="198" relation="straight">
</instruction>
<instruction point="9" id="200" relation="straight">
</instruction>
<instruction point="10" id="202" relation="right">
<landmark id="ns:b3180Door" type="ns:door" relation="lm&gt;"/>
</instruction>
</Direction>
<Direction>
<instruction point="11" id="203" relation="turn ahead">
</instruction>
</Direction>

</ROUTE>
```

origin

destination

Turn around and go straight until the first corridor at your left.

Turn left and go straight until door B3180 that is at your right.

graph nodes **low-level instructions** **segments** **high-level instructions**

Fig. 4. *Sample route with high-level instructions derived from applying Algorithm 1.*

```
:|actor| ( hearer / |person| )
   :|space#direction| (sd / |space#GeneralizedLocation|
      :|space#hasSpatialModality| (lp / |space#LeftProjection| ) ) )
```

This semantic representation expresses a simple turning action to the left. A small subset of possible realisations are (1)-(5) below, which differ along several dimensions, such as the choice of speech function (imperative versus declarative), tense (present versus present continuous), the phoricity of the direction attribute (PP versus AP), or whether or not to use ellipsis or the exact choice of the verb.

(1) "Turn left."
(2) "Turn to the left."
(3) "You are turning left."
(4) "Left."

(5) "Go left."

Under the pCRU framework, we formalise the above variation in a context-free grammar (CFG) consisting of a set of terminal symbols W, a set of nonterminal symbols N, a start symbol S with $S \in N$ and a set of production rules R of the form $n \rightarrow \alpha$, with $n \in N$, $\alpha \in (W \cup N)^*$ and W and N being disjoint. This leads to the following CFG for a `TurningSimple` action.

```
TurningSimple = CONFIGTYPE PROCESS ACTOR SPEECHFUN TENSE DIR (0.7)
TurningSimple = CONFIGTYPE PROCESS ACTOR SPEECHFUN TENSE ":ellipsis full" DIR (0.3)
CONFIGTYPE = "|space#NonAffectingOrientationChange|" (1.0)
PROCESS = ":lex turn" (0.8)
PROCESS = ":lex go" (0.2)
ACTOR = "( hearer / |person| )" (1.0)
TENSE = ":tense present" (0.9)
TENSE = ":tense present-continuous" (0.1)
SPEECHFUN = ":speechact command" (0.9)
SPEECHFUN = ":speechact assertion" (0.1)
DIR = :|space#route| (gr / |space#GeneralizedRoute|
     :|space#direction| (sd / |space#GeneralizedLocation| :phoric-q phoric
          :|space#hasSpatialModality| (sm / LOCATION-DIRECTION ) ) (0.7)
DIR = :|space#route| (gr / |space#GeneralizedRoute|
     :|space#direction| (sd / |space#GeneralizedLocation| :phoric-q notphoric
            :|space#hasSpatialModality| (sm / LOCATION-DIRECTION ) ) (0.3)
```

This representation allows us to capture all arising variation within a single formalism as well as control the application of the respective expansion rules by attaching probabilities to them which indicate each rule's probability of application.

3 Dialogue system evaluation

This evaluation aimed to investigate the performance of our text-based approach for indoor wayfinding. For such a purpose, the dialogue system described above was implemented and tested with a set of users in a real building. This building is complex to navigate; although it has several floors, only one floor was tested.

3.1 Evaluation methodology

We evaluated our dialogue system using objective and subjective metrics mostly derived from the PARADISE framework [26]. This framework is commonly used for assessing the performance of spoken dialogue systems, and can be used for evaluating dialogue systems with different modalities in the wayfinding domain.

The groups of quantitative metrics are described as follows. First, the group of *dialogue efficiency* metrics includes 'system turns', 'user turns', and 'elapsed time' (in seconds). The latter includes the time used by both conversants, from the first user utterance until the last system utterance. Second, the group of *dialogue quality* metrics consists of percentages of parsed sentences, sentences with spotted keywords, and unparsed sentences. Third, the group of *task success* metrics includes the typical binary task success expressed as

$$\text{BinaryTaskSuccess} = \begin{cases} 1 \text{ for finding the target location} \\ 0 \quad\;\; \text{otherwise.} \end{cases} \tag{2}$$

In this group we proposed two additional metrics in order to penalize the degree of difficulty in wayfinding. The first is referred to as '3-valued Task Success (TS)' defined as

$$\text{3-ValuedTS} = \begin{cases} 1 & \text{for finding the target location} \\ 1/2 & \text{for finding the target location with small-medium problems} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

and the second is referred to as '4-valued task success' defined as

$$\text{4-ValuedTS} = \begin{cases} 1 & \text{for finding the target location} \\ 2/3 & \text{for finding the target location with small-medium problems} \\ 1/3 & \text{for finding the target location with severe problems} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The value of 1 is given when the user finds the target location without hesitation, the value with small-medium problems is given when the user finds the location with slight confusion(s), and the value with severe problems is given when the user gets lost but eventually finds the target location. Finally, the group of quantitative metrics are described in Table 3. The sum of scores from these metrics represents the overall user satisfaction score.

3.2 Experimental setup

Our experiments evaluated the dialogue system described above with a user population of 26 native speakers of German. They were university students (16 female, 10 male) aged 22.5 on average. Each user was presented with six wayfinding tasks, resulting in a total of 156 route dialogues. They were asked in each case to find a particular location based on the route instruction generated by the dialogue system on request by the user. The locations were spatially distributed. Two tasks used 2 High-Level Instructions (HLIs), two tasks used 3 HLIs, and two tasks used 4 HLIs. The dialogue tasks were executed pseudorandomly (from a uniform distribution). At the beginning of each session, participants were asked about their familiarity with the building using a 5-point Likert scale, where 1 represents the lowest familiarity and 5 the highest. This resulted in a familiarity score of 2.4. Then, our participants received the following set of instructions: (a) you can ask the system using natural language, (b) you can take notes from the received instructions, (c) follow the instructions as precisely as possible, (d) you are not allowed to ask anyone how to get to the target location, and (e) you can give up anytime after trying without success by telling that to the assistant that will follow you. At the end of each wayfinding task, participants were asked to fill a questionnaire (Table 3) for obtaining qualitative results using a 5-point Likert scale, where 5 represents the highest score.

Table 3. Subjective measures for evaluating indoor wayfinding, adapted from [26].

Measure	Question
Easy to Understand	Was the system easy to understand?
System Understood	Did the system understand what you asked?
Task Easy	Was it easy to find the location you wanted?
Interaction Pace	Was the pace of interaction with the system appropriate?
What to Say	Did you know what you could write at each point?
System Response	Was the system fast and quick to reply to you?
Expected Behaviour	Did the system work the way you expected it to?
Future Use	Do you think you would use the system in the future?

3.3 Experimental results

According to dialogue efficiency metrics, it can be observed from Table 4 that the user-machine interactions involved short dialogues in terms of system turns, user turns and time. These results suggest that once users receive instructions to find a given location, they tend not to ask further questions. We can also observe a large number of words per system turn mostly due to the textual instructions, where the longer the number of high-level instructions the longer the textual output. In addition, although some users used only keywords in the textual input, overall they asked questions.

According to dialogue quality, it can be noted that our grammars did not have wide coverage. There are many different ways to ask for a given location, including sentences with ungrammatical structures and sentences with words absent in the lexicon. However, the keyword spotter then was crucial for identifying the users' target location.

According to task success, our dialogue system obtained a very high binary task success, but this measure does not take into account how hard it was for the user to find the given locations. In contrast, whilst our 3-valued task success measure penalizes more strongly, our 4-valued task success measure is between the other two metrics. From these metrics, we found that the latter generated more faithful scores because it predicts more closely user satisfaction. This argument can be validated with statistical analysis, but this is left as future work.

Our qualitative results report very high scores for user satisfaction, mainly for the dialogues with 2 High-Level Instructions (HLIs) and 3 HLIs. However, users found it harder to follow the dialogues with 4 HLIs. One can think that the reason was due to the length of the instructions, but we observed that it was more due to ambiguity in which corridors to follow. The lower scores in the following qualitative metrics support this argument: easy to understand, task easy, expected behaviour and future use. Nevertheless, we found that a dialogue system for indoor wayfinding using language processing capabilities — with only text input and output — can obtain very high overall scores in user satisfaction.

Table 4. Average results of our wayfinding system for dialogues with different amounts of High-Level Instructions (HLIs), organized according to the following groups of metrics: dialogue efficiency, dialogue quality, task success and user satisfaction.

Measure	2 HLIs (52 dialogues)	3 HLIs (52 dialogues)	4 HLIs (52 dialogues)	All (156 dialogues)
Avg. System Turns	2.25	2.38	2.28	2.30
Avg. User Turns	1.30	1.61	1.64	1.52
Avg. System Words per Turn	34.05	40.04	49.59	41.30
Avg. User Words per Turn	4.06	5.34	4.84	4.79
Avg. Time (in seconds)	20.69	19.77	25.87	22.14
Parsed Sentences (%)	23.8	4.3	22.5	16.7
Spotted Keywords (%)	74.6	91.4	73.2	79.9
Unparsed Sentences (%)	1.6	4.3	4.2	3.4
Binary Task Success (%)	96.2	100.0	88.5	94.9
3-Valued Task Success (%)	92.3	88.5	63.5	81.4
4-Valued Task Success (%)	94.9	92.3	75.6	87.6
Easy to Understand	4.65	4.6	4.08	4.46
System Understood	4.71	4.62	4.62	4.65
Task Easy	4.60	4.54	3.73	4.29
Interaction Pace	4.71	4.65	4.52	4.63
What to Say	4.71	4.63	4.65	4.66
System Response	4.60	4.62	4.58	4.56
Expected Behaviour	4.64	4.50	4.21	4.45
Future Use	4.46	4.37	4.12	4.31
User Satisfaction (sum)	37.1	36.5	34.5	36.0
User Satisfaction (%)	92.7	91.2	86.3	90.0

Finally, we included an additional question in the survey filled after each dialogue: ‘Did you find the location only based on the given instructions by the system or did you use additional help such as signs?’ This question also used a 5-point Likert scale, where 5 represents the highest score for strictly following only the system instructions. This resulted in an average value of 4.3, which suggests that the results described above were derived from following almost entirely the system’s instructions.

4 Conclusions and future work

In this paper we have presented a dialogue system for indoor wayfinding in a complex building using text-based natural language input and output. The system was described with four main components: natural language understanding, dialogue management, route instruction generation and natural language generation. In the latter we described our approach based on high-level instructions. A key advantage of our dialogue system is its support for language-independence, only parsing and generation grammars have to be added in order to support a new language, the rest is reused. Experimental results — using the PARADISE evaluation framework — in a real environment with 26 participants (156 dialogues) provide evidence to support the following claims: (a) text-based dialogues resulted in very short interactions, they mostly consist of question and answer, though eventually clarifications or apologies occurred; (b) keyword spotting was an essential component to assist the parser with unparsable utterances; (c) our proposed 4-valued task success metric predicts better user satisfaction than binary task success or 3-valued task success; and (d) a text-based dialogue system for indoor wayfinding can obtain very high overall scores in user satisfaction. To the best of our knowledge, this is the first evaluation of its kind in the indoor wayfinding domain.

We suggest the following avenues for future research:

First, text-based language processing, spoken language processing and graphical interfaces (such as maps) can be combined into principled frameworks for building effective wayfinding systems. Such systems should be evaluated as in this paper in order to assess the performance across different system versions. In this way, systematic evaluations can be made by varying different conditions under a given framework. This is an important and useful step to take that has not so far been achieved in indoor navigation.

Second, the dialogue manager is responsible for controlling the system's dialogue behaviour. When the system's behaviour becomes complex, it is less recommendable to use hand-crafted behaviour because it is non-adaptive and labour intensive. Machine learning methods such as reinforcement learning can be used to induce the system's behaviour automatically [27, 17, 28]. This is relevant for learning adaptive and complex behaviour such as learning to ground, learning to clarify, learning to present information, learning multimodal strategies and learning to negotiate route directions.

Third, in the case of indoor route directions, future work can entail covering paths that cross multiple floors. This will require both handling a graph with dedicated transition nodes between floors and a clear communication of floor changes in the route directions. In the present work, we used corridors as main landmarks; however, a principled mechanism to rank indoor landmarks can be investigated. In addition, providing route instructions for new spatial environments is possible by providing spatial representations of additional environments in the form of a route graph.

Finally, future work in language generation can aim to enhance the adaptiveness of route descriptions along three dimensions: (a) to make descriptions more tailored towards a particular user by taking their familiarity of the environment into closer consideration [29]; (b) to present information for users with different cognitive styles for users familiar or unfamiliar with a given environment [30, 31, 32]; and (c) to investigate how to incorporate interactive alignment [33].

References

[1] Golledge, R.: Wayfinding behavior: Cognitive mapping and other spatial processes. Johns Hopkins Press, Baltimore, MD (1999)

[2] Passini, R.: Wayfinding: A conceptual framework. Urban Ecology **5**(1) (1981) 17–31

[3] Dale, R., Geldof, S., Prost, J.P.: Using natural language generation in automatic route description. Journal of Research and Practice in Information Technology **37**(1) (2005) 89–105

[4] Kray, C., Kortuem, G., Krüger, A.: Adaptive navigation support with public displays. In Amant, R.S., Riedl, J., Jameson, A., eds.: Proceedings of IUI 2005. ACM Press, New York. (2005) 326–328

[5] Callaway, C.: Non-localized, interactive multimodal direction giving. In van der Sluis, I., Theune, M., Reiter, E., Krahmer, E., eds.: Proceedings of the Workshop on Multimodal Output Generation MOG 2007, Centre for Telematics and Information Technology (CTIT), University of Twente (2007) 41–50

[6] Münzer, S., Stahl, C.: Providing individual route instructions for indoor wayfinding in complex, multi-level buildings. In Probst, F., Keßler, C., eds.: Proceedings of the 5th Geographic Information Days, Münster, IfGIprints (2007) 241–246

[7] Hochmair, H.H.: Pda-assisted indoor-navigation with imprecise positioning: Results of a desktop usability study. In Meng, L., Zipf, A., Winter, S., eds.: Map-based Mobile Services - Interactivity, Usability and Case Studies. Springer, Berlin, Heidelberg (2008) 228–247

[8] Becker, T., Nagel, C., Kolbe, T.H.: A multilayered space-event model for navigation in indoor spaces. In Lee, J., Zlatanova, S., eds.: Proceedings of the 3rd International Workshop on 3D Geo-Information, Seoul, Korea, Berlin, Heidelberg, Springer (2008)

[9] Ohlbach, H.J., Stoffel, E.P.: Versatile route descriptions for pedestrian guidance in buildings: Conceptual model and systematic method. In: 11th AGILE International Conference on Geographic Information Science, University of Girona, Spain (2008)

[10] Kruijff, G.J.M., Zender, H., Jensfelt, P., Christensen, H.I.: Situated Dialogue and Spatial Organization: What, Where... and Why? International Journal of Advanced Robotic Systems **4**(1) (March 2007) 125–138 Special Issue on Human-Robot Interaction.

[11] Clark, S., Hockenmaier, J., Steedman, M.: Building Deep Dependency Structures using a Wide-Coverage CCG Parser. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). (2002) 327–334

[12] Bateman, J.A.: Enabling technology for multilingual natural language generation: the KPML development environment. Journal of Natural Language Engineering **3**(1) (1997) 15–55

[13] Ross, R.J., Bateman, J.A.: Daisie: Information state dialogues for situated systems. In: TSD. Volume 5729 of LNCS., Springer (2009) 379–386

[14] Collier, R.W.: Agent Factory: A Framework for the Engineering of Agent Oriented Applications. PhD thesis, University College Dublin (2001)

[15] Bateman, J., Hois, J., Ross, R., Tenbrink, T., Farrar, S.: The Generalized Upper Model 3.0: Documentation. SFB/TR8 internal report, Collaborative Research Center for Spatial Cognition, University of Bremen, Germany (2008)

[16] Lemon, O., Georgila, K., Henderson, J.: Evaluating efectiveness and portability of reinforcement learned dialogue strategies with real users: The TALK TownInfo evaluation. In: SLT, Palm Beach, Aruba (Dec 2006) 178–181

[17] Cuayáhuitl, H.: Hierarchical Reinforcement Learning for Spoken Dialogue Systems. PhD thesis, School of Informatics, University of Edinburgh (January 2009)

[18] Richter, K.F.: Context-Specific Route Directions - Generation of Cognitively Motivated Wayfinding Instructions. Volume DisKi 314 / SFB/TR 8 Monographs Volume 3. IOS Press, Amsterdam, The Netherlands (2008)

[19] Klippel, A.: Wayfinding choremes. In Kuhn, W., Worboys, M., Timpf, S., eds.: Spatial Information Theory. Foundations of Geographic Information Science, Berlin, International Conference COSIT, Springer (2003) 320–334 LNCS 2825.

[20] Hansen, S., Richter, K.F., Klippel, A.: Landmarks in OpenLS - a data structure for cognitive ergonomic route directions. In Raubal, M., Miller, H., Frank, A.U., Goodchild, M.F., eds.: Geographic Information Science - Fourth International Conference, GIScience 2006, Springer; Berlin (2006) 128–144 LNCS 4197.

[21] Richter, K.F.: A uniform handling of different landmark types in route directions. In Winter, S., Duckham, M., Kulik, L., Kuipers, B., eds.: Spatial Information Theory. LNCS 4736, Springer; Berlin (2007) 373–389 COSIT.

[22] Klippel, A., Tappe, H., Kulik, L., Lee, P.U.: Wayfinding choremes — a language for modeling conceptual route knowledge. Journal of Visual Languages and Computing **16**(4) (2005) 311–329

[23] Richter, K.F., Duckham, M.: Simplest instructions: Finding easy-to-describe routes for navigation. In Cova, T.J., Miller, H.J., Beard, K., Frank, A.U., Goodchild, M.F., eds.: Geographic Information Science - 5th International Conference, GIScience 2008. LNCS 5266, Springer; Berlin (2008) 274–289

[24] Belz, A.: Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. Natural Language Engineering **1** (2008) 1–26

[25] Kasper, R.: SPL: A Sentence Plan Language for text generation. Technical report, USC/ISI (1989)

[26] Walker, M., Kamm, C., Litman, D.: Towards developing general models of usability with PARADISE. Natural Language Engineering **6**(3) (2000) 363–377

[27] Singh, S., Litman, D., Kearns, M., Walker, M.: Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. JAIR **16** (2002) 105–133

[28] Cuayáhuitl, H., Renals, S., Lemon, O., Shimodaira, H.: Evaluation of a hierarchical reinforcement learning spoken dialogue system. Computer Speech and Language **24**(2) (2010) 395–429

[29] Tenbrink, T., Winter, S.: Variable Granularity in Route Directions. Spatial Cognition and Computation **9** (2009) 64–93

[30] Lovelace, K.L., Hegarty, M., Montello, D.R.: Elements of good route directions in familiar and unfamiliar environments. In: COSIT '99: Proceedings of the International Conference on Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, London, UK, Springer-Verlag (1999) 65–82

[31] Burnett, G., Smith, D., May, A.: Supporting the navigation task: characteristics of good landmarks. In: Proceedings of the Annual Conference of the Ergonomics of Society. Turnin, Italy. (2001) 441–446

[32] May, A.J., Ross, T., Bayer, S.H.: Driver's Information Requirements when Navigating in an Urban Environment. Journal of Navigation **56** (2003) 89–100

[33] Pickering, M.J., Garrod, S.: Toward a mechanistc psychology of dialog. Behavioral and Brain Sciences **27** (2004) 169–226