A Classification-Based Approach to Automating Human-Robot Dialogue

Felix Gervits, Anton Leuski, Claire Bonial, Carla Gordon, and David Traum

Abstract We present a dialogue system based on statistical classification which was used to automate human-robot dialogue in a collaborative navigation domain. The classifier was trained on a small corpus of multi-floor Wizard-of-Oz dialogue including two wizards: one standing in for dialogue capabilities and another for navigation. Below, we describe the implementation details of the classifier and show how it was used to automate the dialogue wizard. We evaluate our system on several sets of source data from the corpus and find that response accuracy is generally high, even with very limited training data. Another contribution of this work is the novel demonstration of a dialogue manager that uses the classifier to engage in multi-floor dialogue with two different human roles. Overall, this approach is useful for enabling spoken dialogue systems to produce robust and accurate responses to natural language input, and for robots that need to interact with humans in a team setting.

F. Gervits (✉)
Tufts University, Medford, MA, USA
e-mail: felix.gervits@tufts.edu

A. Leuski · C. Gordon · D. Traum
USC Institute for Creative Technologies, Playa Vista, CA, USA
e-mail: leuski@ict.usc.edu

C. Gordon
e-mail: cgordon@ict.usc.edu

D. Traum
e-mail: traum@ict.usc.edu

C. Bonial
U.S. Army Research Laboratory, Adelphi, MD, USA
e-mail: claire.n.bonial.civ@mail.mil

E. Marchi et al. (eds.), *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, Lecture Notes in Electrical Engineering 714,
https://doi.org/10.1007/978-981-15-9323-9_10

1 Introduction

A major drive in human-robot interaction (HRI) research is to enable robots to serve as genuine partners in teams with humans. Such heterogeneous teams are intended for use in a variety of applications including classroom tutoring, disaster-relief, and military reconnaissance. In particular, there has been a great deal of research in task-oriented remote communication for the purpose of urban search and rescue (USAR). HRI is desirable for these domains, as a robot can be used to explore a hazardous area while a human monitors the situation and gives instructions remotely.

A critical requirement of effective teaming in collaborative USAR domains is communication [7, 19]. Humans use communication to share task-relevant information, give instructions, discuss plans, and many other functions. As a result, robots will need to handle at least some of these functions if they are expected to fill the role of a human teammate. At minimum, robots will need to interpret a command in the form of speech input, perform the corresponding action of the command, and produce a feedback response to the human. This involves bidirectional communication in which the robot not only takes orders but also responds in meaningful ways. Error handling and dialogue management are additional requirements needed for more robust interactions. Finally, naturalness and flexibility of the system are also desirable: it is important that humans can talk to the robot in a natural manner, which includes all the disfluencies and irregularities that arise in natural language (NL), and the robot should be robust to variability in speech in order to serve as a more effective conversational partner.

1.1 Background and Related Work

In order to meet the above requirements, various types of dialogue models have been proposed and attempted. The most basic is finite-state systems in which dialogues are represented as a pre-determined state transition network [18]. Finite-state systems are effective for small, highly-structured domains in which the flow of dialogue is known in advance. However, such systems are generally inflexible to input that is not in the network, and do not seem well suited to the complex USAR domains of interest. Frame-based dialogue models have also been proposed, which involve filling in various slots in a "form" corresponding to an action or utterance [26]. These offer more flexibility to handle increasingly complex dialogues, but struggle with utterances that do not fit into a frame. As a result, frame-based models have been mainly used in tasks with a fixed set of slots, such as travel booking [8]. Finally, plan-based systems turn dialogue into a planning problem in which a human's utterance is mapped to a speech act and the system performs logical inference over its beliefs and goals in order to select an appropriate response [1, 4]. While this kind of model is very useful for handling complex dialogues, it relies on the difficult problem of identifying speech acts and intentions.

We adopt the *corpus-based robotics* approach, wherein the system is trained on corpus data from the target domain [3]. The corpus used for training could involve human-human instruction [6], human-robot instruction [17], or human-robot instruction in a Wizard-of-Oz (WoZ) paradigm [2, 5]. The corpus data includes examples of natural commands that robots will need to interpret and act on, and serves as a source of interaction patterns to inform dialogue management policies. Through the use of statistical techniques, systems using this approach have been very effective at modeling various aspects of dialogue [20, 24, 25]. This approach also offers flexibility, as data-driven models are often robust to noisy and disfluent data. However, a major drawback with many machine learning techniques is that large, annotated data sets are required for training [21]. Such data sets are often unavailable or infeasible to produce due to the large cost and effort needed to collect, transcribe, and annotate data. Moreover, new ones are often needed for each target domain because the systems do not usually extend beyond the particular training domain.

1.2 Motivation and Present Work

We are currently developing an end-to-end spoken dialogue system for use in a collaborative human-robot navigation domain. The system is trained on a small corpus involving a dual WoZ setup in which one wizard handles the dialogue management (DM) and the other handles robot navigation (RN). The rationale behind using such a corpus is that we wanted the system to interpret speech and respond in an appropriate, human-like manner. This approach provides data-driven insights into what such a response would be, and what variety we should expect, in the context of a collaborative navigation task. Our ultimate goal is to create a fully autonomous robot. In this paper we describe initial attempts to automate the natural language dialogue capabilities using a statistical classifier based on cross-language information retrieval. The system operates across multiple floors (i.e., distinct communication channels) and "translates" messages between the human user and the RN component or wizard, and gives positive and negative feedback to the human user.

Given our small corpus, we were interested in exploring how far we can get with a data-driven approach using such limited training data and limited annotation. Most end-to-end systems require large training sets to get reasonable performance, but previous evaluations of a similar classifier have shown reasonably high accuracy with only a few hundred utterances for training [9] compared to the hundreds of thousands needed in other systems (e.g., [20]). Note that we do not claim that our approach is immune to the limitations of other data-driven systems, and we discuss some of these limitations in Sect. 5. However, the goal is to mitigate some of these limitations through our classification and DM approach.

Below, we introduce our task domain and provide details of the corpus used. Next, we describe our classification approach as well as the DM policies that were implemented. Finally, we evaluate our system on several data sets of varying size from the corpus to compare response accuracy. In the evaluation, the following points

will be addressed: (1) accuracy of the classifier (especially as it relates to the size and composition of the training data), (2) adequacy of the DM response, and (3) integration of the system in a robotic architecture.

2 Collaborative Human-Robot Navigation Task

2.1 Task Domain

Our task domain involves collaborative navigation akin to a USAR scenario. In the task, a human serves as a *Commander* and supervises a remotely-located robot to perform a navigation task in an unfamiliar physical environment. The environment is modeled after a house and includes various rooms and objects consistent with this environment type (rooms, hallways, etc.). The goal of the task is to work together as a team to accomplish two subtasks—one related to searching (e.g., locate shoes) and one related to analysis (e.g., evaluate whether the area can serve as a headquarters).

Throughout the task, the Commander is seated in front of a computer with an interface showing task-relevant information. The interface includes a 2D occupancy grid showing the robot's location, a snapshot of the last image taken by the robot, and a textbox showing the robot's dialogue responses (see top-right of Fig. 1). To direct the robot, the Commander is able to speak freely using unconstrained natural language. Examples of common instructions include "Move forward 10 feet", "Take a picture", and "Turn right 45°". People also used landmark-based instructions such as "Move to face the yellow cone", and "Go to the doorway to your right", although these were less common than the metric-based instructions [15].

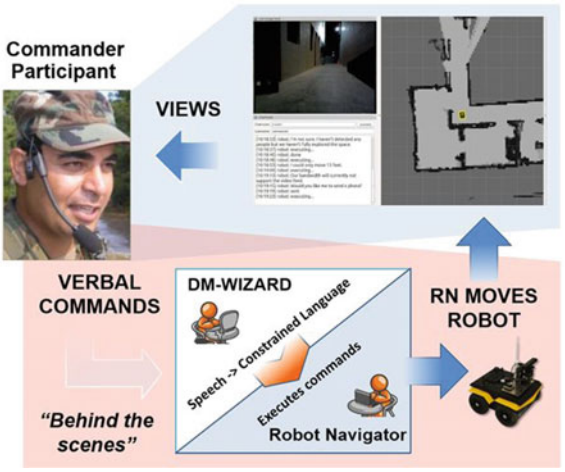

Fig. 1 Experimental task domain with dual-wizard setup (from [13])

The task was run using a dual-WoZ setup wherein one wizard controlled the DM and the other controlled the RN. Importantly, the wizards had to communicate with one another to ensure that actions and responses were performed correctly and in a timely manner [14]. The task was run over several experiments, with additional experiments currently in progress. In Experiment (Exp.) 1, the DM-Wizard typed free responses to the Commander and RN-Wizard according to pre-established guidelines (see [13]). From this, we developed a GUI that was used by the DM-Wizard in Exp. 2 to provide quicker and more uniform responses [2, 16]. The same GUI was used in Exp. 3, except that here we used a simulated robot and environment rather than a physical one. Exp. 1 and 2 had 10 participants each, whereas Exp. 3 had 62 participants.

2.2 *Corpus and Annotation*

A corpus was created from the Exp. 1 and 2 data (annotation for Exp. 3 is still in progress). Dialogues were annotated according to the scheme described in [23], which was specifically designed to handle the multiple conversational floors in our dual-wizard setup. These floors include: (1) Commander and DM-Wizard and (2) DM and RN-Wizards. The main unit of dialogue in our annotation scheme is the *transaction unit (TU)*, which includes the initial utterance expressing the intent of the speaker and all subsequent utterances across all floors that are used to achieve the intent of the original speaker. An example TU can be found in Table 1. In addition, our scheme also includes three distinct types of *relations*, which are used to characterize

Table 1 Example TU and annotation from the corpus. The * indicates that the antecedent is part of a sequence of expansions

	Left floor		Right floor		Annotation	
#	Commander	DM ->Commander	DM ->RN	RN	Ant.	Rel.
1	Rotate to the right ninety degrees					
2	And take a photo				1	Continue
3		Ok			2*	Ack-understand
4			Turn right 90°		1	Translation-r
5			Then...		4	Link-next
6			Send image		2	Translation-r
7				Done and sent	6*	Ack-done
8		Done, sent			7	Translation-l

how an utterance is related to an antecedent (previous utterance). These relation types include *expansions*, *responses*, and *translations* along with various subtypes of each. Expansions are continuations of a previous utterance by the same speaker in the same floor. Responses are produced by different speakers in the same floor, and include several types of acknowledgements, clarifications, and answers. Finally, translations are used to relate utterances in different floors, and include two subtypes: *translation-right* involves the DM-Wizard translating a Commander's instruction to the RN-Wizard for action (e.g., "Move forward three feet"), and *translation-left* involves the DM-Wizard translating the RN-Wizard's action to the Commander in the form of feedback (e.g., "I moved forward three feet"). In total, the corpus included 2230 TUs across 60 dialogues from 20 different Commanders (each Commander participated in three dialogues) [23].

3 Natural Language Dialogue Processing

In this section, we provide an overview of the NL approach toward mimicking the DM-Wizard's utterance selection policies based on input from Commander instructions. We first outline the classification approach and describe the data processing that we carried out on the corpus data. We then describe the DM policies that were implemented to make use of the classifier output in order to produce appropriate responses across the multiple floors. Finally, we evaluate the output on new test data from Exp. 3.

3.1 Classifier Approach

The task of the language classifier is to analyze the Commander's instruction and select the appropriate utterances from the system's collection of responses. It involves the following three step process:

First, the classifier indexes the existing language data—a dataset of instruction-response pairs that we have collected during WoZ experiments. It generates a statistical language model for each natural language utterance (both instruction and response): $P(w|W)$, where w is a word in the vocabulary and W is the utterance. Note that the vocabularies of instructions and responses are different.

Next, the classifier uses the indexed data to construct a joint model of the instructions and responses, and to compute the cross-language relevance model for each new Commander's instruction $P(w|C)$, where w is a word in the response vocabulary and C is the instruction. Please see our previous paper [11] for the technical details of the approach.

Finally, the classifier compares the language model $P(w|C)$ with the language model of each response in the system's dataset, $P(w|R_i)$. Here R_i is the ith response

in the dataset. It returns all the responses with the similarity score above a predefined threshold. The threshold is determined during the classifier training phase.

The classifier implementation is part of the NPCEditor platform, which has been used in the past to build effective question-answering conversational systems [10]. The approach requires a relatively small amount of training data, has a small number parameters to tune (three parameters, including the threshold, in most cases), and is robust to noise and errors in the input [9]. Next, we explain how we processed our experimental data to train the classifier.

3.2 Data Processing

Instruction-Response pairs The first step was to constrain the multi-floor data to something closer to what the classifier uses in terms of linked initiative-response pairs. This is challenging because in our data there are two different types of reactions to a Commander input: *responses* to the Commander (including positive and negative feedback) and *translations* of actionable Commander instructions to the RN. To create this dataset, we first used a script to parse the annotated corpus data and link each utterance produced by the Commander with the DM-Wizard's responses to it. We did this for several relation types, including translation-right, and several response subtypes (clarification, acknowledgment, answer, etc.); this resulted in a set of instruction-response pairs. For example, "Take a picture" → "image" is an example of a *translation-right* pair, in which a Commander's instruction is translated into a shorthand request sent to the RN-Wizard, and "Move forward" → "Please tell me how far to move forward" is an example of a *request-clarification* pair, in which the Commander's open-ended instruction prompts a clarification request from the DM-Wizard.

Coherence rating Since the resulting set of instruction-response pairs were automatically generated from scripts, the next step involved filtering these pairs for coherence. We used the 4-point coherence rating scale from [22], where a 1 represents a response that is either missing or irrelevant to the instruction, a 2 represents a response that relies on external context to match, a 3 represents a response that indirectly addresses the instruction but that contains additional (irrelevant) information, and a 4 represents a response that directly addresses the instruction. Using this rating scale, we manually inspected the instruction-response pairs from Exp. 1 and 2 and rated each one. In Exp. 1, out of a total of 999 pairs, 96 pairs had a rating of 1, 222 pairs had a rating of 2, 1 pair had a rating of 3, and 680 pairs had a rating of 4. In Exp. 2, out of a total of 1419 pairs, 50 pairs had a rating of 1, 387 pairs had a rating of 2, 4 pairs had a rating of 3, and the remaining 978 pairs had a rating of 4. For the final training set, we included all the pairs that had a 3 or 4 coherence rating.

Data smoothing Finally, we conducted a smoothing step in order to ensure that the training data would cover the various fields in the most common instructions. For example, a command such as "Move forward four feet" might not exist in the corpus,

but is nonetheless an instruction that the system should be able to carry out. In order to capture these missing fields, we added a set of 250 pairs to the training data. 196 of these "smoothing pairs" came from the table which was used to generate the Exp. 2 GUI (see [2]). This ensured that the system could at least interpret the most common actionable commands from the experimental studies. The remaining 54 pairs were hand-generated to fill in values that were missing from the corpus data, and simply included additional values for the existing commands.

3.3 Dialogue Manager Policies

After being trained on the instruction-response pairs as described above, the classifier learned a mapping between commands and responses from the data. We then implemented a DM whose role it was to use the classifier output to select appropriate responses and send them to the corresponding floor (Commander or RN).

The DM works in the following manner. First, it receives an utterance in the form of a string after it has passed through the speech recognizer. The classifier then ranks the top responses that match the instruction and sends this list back to the DM. Upon receiving the matching response from the classifier, the DM then sends this to the corresponding floor. Actionable commands are formatted and sent to the RN whereas the corresponding feedback message ("Executing", "Moving", "Turning", etc.) is sent to the Commander's interface. Non-actionable commands cause the system to generate a response (usually a type of clarification or acknowledgment) to the Commander in order to repair the instruction.

Some specific policies were implemented to handle problematic input. One such policy handles the case when the classifier finds no match. This usually means that the command was outside of the domain, or that any potential matches were below threshold. In either case, the DM will cycle through several general clarification requests when this happens, prompting the Commander to repeat and reformulate the instruction. Another policy was implemented to handle cases in which the classifier selected multiple responses. In this case, it always picks the one with the highest score, but in the case of a tie, a random response is chosen from the tied options.

4 Evaluation

The DM in combination with the trained classifier allowed us to replicate many of the dialogue behaviors from the experimental data. To evaluate the performance of our system, we trained six classifiers using varying amounts of source data from each of the first two experiments (Exp. 1 and Exp. 2). Training data for each classifier consisted of annotated user utterances from a given experiment, which were processed using the methods described in Sect. 3.2. Additional smoothing data was added to each of the three original classifiers in order to test for the benefit of includ-

ing these extra pairs. A summary of the combinations of data, as well as the number of training utterances, responses and links between the two is presented in Table 2. The "# links" column refers to the number of connections between utterances and responses in that set. These connections do not represent a one-to-one relationship, as any given utterance can be linked to more than one response, and vice versa. We also evaluated the DM separately from the classifier to test the appropriateness of responses produced by the system.

4.1 Classifier Evaluation and Results

Each of the six trained classifiers was tested on a test set comprised of three previously unseen dialogues that were randomly selected from Exp. 3. These dialogues were annotated, and the instruction-response pairs were extracted, but no other processing was done on these pairs as we sought to maintain the raw data for testing. In total, the test set included 183 instruction-response pairs.

For each utterance in the test set, we compared the best classifier match to the expected output, which is the one actually produced by the DM-Wizard in the test data. Accuracy was calculated as the percentage of queries where the best classifier match is the expected response. The results of our evaluation are displayed in the right-most column of Table 2. Accuracy scores ranged from 61% in the Exp. 2 classifier to 75% in the combined Exp. 1+2 classifier with added smoothing pairs. In general, we found that performance improved with the addition of the smoothing pairs, and this improvement ranged from 2.7 to 5.5% depending on the original training set. As expected, this largely benefited the Exp. 1 and Exp. 2 classifiers, which had limited data and were missing many of the basic pairs that were part of the smoothing set. Interestingly, we found that accuracy on the unconstrained Exp. 1 data (65%) was higher than the GUI-based Exp. 2 data (61%). This is likely due to the reduced number of responses in Exp. 2 caused by the standardization of the GUI. Without the smoothing data added, there may not have been enough unique responses to match the test queries. Overall, the highest accuracy (75%) was obtained for the classifier trained on all the data. This is a promising result, and suggests that relatively high accuracy can be achieved with under 1000 utterances of training.

4.2 Dialogue Adequacy Evaluation

It is important to note that classifier accuracy is only part of what we are interested in. Perfect matches are of course desirable, however a response can still be reasonably appropriate even if not an exact match of the corpus data (e.g., "turn 20°" vs "turn 25°"). In order to evaluate the classifier in terms of expected impact on the dialogue, we examined the 45 (about 25% of test set) utterances that the combined classifier got wrong in the previous evaluation. For each of these responses, we placed them into

Table 2 Classifier data summary. Accuracy represents the proportion of the classifier's responses that matched the test query

Training data	Test data	#Utterances	#Responses	#Links	Accuracy
Exp. 1 only	Exp 3	347	247	366	0.6503
Exp. 1 + smoothing	Exp 3	593	436	614	0.6831
Exp. 2 only	Exp 3	424	141	429	0.6066
Exp. 2 + smoothing	Exp 3	670	328	675	0.6612
Exp. 1 & 2	Exp 3	722	303	751	0.7268
Exp. 1 & 2 + smoothing	Exp 3	966	483	995	**0.7541**

Table 3 Dialogue adequacy evaluation showing the type and relative frequency of the 45 system responses that did not match the test set

	Felicitous	Approximate	Context-dependent	Wrong	No response
Instruction	Turn one eighty	Go west five feet	Go to plant	Go back to table	Rotate toward camera towards calendar
Test-set response	Turn 180	Turn to face west; move forward 5 feet	Go to Dark room plant	Move back towards table	Move to conference calendar
DM response	Rotate 180	Turn to face west; move forward 10 feet	Go to Foyer plant	Return to starting point	< no response >
Count (out of 45)	8	15	14	7	1
Proportion	0.18	0.33	0.31	0.16	0.02

one of five categories: *Felicitous*—appropriate responses that would have the same effect as the correct response, *Approximate*—responses that differed only slightly from the correct one (e.g., variation in turn radius or movement distance), *Context-Dependent*—responses that could be correct, but that depend on the context in which they occurred, *Wrong*—responses that were not appropriate for the given command, and *No Response*—indicating that the classifier did not find a match. Table 3 summarizes the analysis of responses that did not match the test-set, including examples of each type and the frequency of each.

Felicitous responses are expected to have no negative impact on the dialogue. Approximate responses might have a small delay to extend or correct the robot's behavior. Wrong responses are expected to have a more severe impact in terms

of either cancelling the instruction mid-operation, or undoing it after. Context-Dependent responses might either have no negative effect (like Felicitous responses) or a negative effect (like Wrong responses), depending on how the context is applied to create a full instruction. When the classifier does not find a match, the DM instructs the user to repeat or rephrase the previous instruction, slowing down the dialogue, but not impacting the robot's physical situation.

In our analysis, we found that over half of the incorrect responses in the test set were either *Felicitous* or *Approximate* to the correct response. This suggests that, despite not matching the test data, these responses would still be appropriate and would advance the dialogue. Only one case had no response, and the remaining cases were split between the *Context-Dependent* and *Wrong* categories. Fortunately, these responses were infrequent, representing only 11% of the total test set.

4.3 Demonstration: Integration in the ScoutBot Architecture

One of the primary goals of this research project is to develop a fully automated end-to-end dialogue system for collaborative navigation. To that end, we (and colleagues) have implemented a system called ScoutBot, which was designed to automate the tasks of the dual wizards in our navigation task [12]. We have found in pilot testing that Scoutbot can effectively interpret simple instructions and navigate accordingly, but a more detailed evaluation is work in progress. Currently, the main limitation of Scoutbot is the inability to handle landmark-based instructions such as "Move from A to B". Addressing this will require additional mechanisms (see below), but importantly, the system still works well for the majority of examples and should be sufficient for the team to complete the task. A demonstration video of ScoutBot can be found at the following link: http://people.ict.usc.edu/~traum/Movies/scoutbot-acl2018demo.wmv.

5 Conclusion and Future Work

The ability to converse with robots is an important part of human-robot teaming envisioned for many applications. As a result, end-to-end dialogue systems that facilitate effective communication are becoming increasingly needed. We presented a data-driven system to achieve this goal that uses a statistical classifier and dialogue manager to interpret natural language commands, produce appropriate responses, and carry out actions. In our evaluation, the system was shown to maintain relatively high response accuracy even with limited and noisy training data.

Moving forward, we are in the process of extending the system to handle some of the limitations we encountered, namely landmark-based instructions and complex, multi-turn commands. The former will require a context model in which the system tracks the robot's location throughout the map and biases the DM to favor objects

and locations in the local context. A possible solution for the latter is supplementing our system with an information extraction approach in which the key parameters of a command (e.g., action, distance, etc.) are extracted and used to fill a semantic frame. This will also enable us to provide more detailed clarification requests to obtain specific pieces of information (e.g., "how far should I move forward?"). Finally, we expect the additional data from Exp. 3 to further improve the classifier accuracy and reduce the number of incorrect responses. Overall, this approach offers a practical alternative to those that require large-scale corpora for dialogue systems, and shows that good performance is possible with a data-driven approach using a smaller data set.

Acknowledgements This research was sponsored by the U.S. Army Research Laboratory and by a NASA Space Technology Research Fellowship for the first author.

References

1. Allen JF, Perrault CR (1980) Analyzing intention in utterances. Artif Intell 15(3):143–178
2. Bonial C, Marge M, Foots A, Gervits F, Hayes CJ, Henry C, Hill SG, Leuski A, Lukin SM, Moolchandani P, Pollard KA, Traum D, Voss CR (2017) Laying down the yellow brick road: development of a wizard-of-oz interface for collecting human-robot dialogue. In: Symposium on natural communication for human-robot collaboration, AAAI FSS
3. Bugmann G, Klein E, Lauria S, Kyriacou T (2004) Corpus-based robotics: a route instruction example. In: Proceedings of intelligent autonomous systems, pp 96–103
4. Cohen PR, Perrault CR (1979) Elements of a plan-based theory of speech acts. Cogn Sci 3(3):177–212
5. Dahlbäck N, Jönsson A, Ahrenberg L (1993) Wizard of oz studies - why and how. Knowl-Based Syst 6(4):258–266
6. Eberhard KM, Nicholson H, Kübler S, Gundersen S, Scheutz M (2010) The indiana "cooperative remote search task" (crest) corpus. In: Proceedings of LREC 2010
7. Gervits F, Eberhard K, Scheutz M (2016) Team communication as a collaborative process. Front Robot AI 3:62
8. Issar S, Ward W (1993) CMLPs robust spoken language understanding system. In: Third European conference on speech communication and technology
9. Leuski A, Traum D (2008) A statistical approach for text processing in virtual humans. In: Proceedings of the 26th army science conference
10. Leuski A, Traum D (2011) NPC editor: creating virtual human dialogue using information retrieval techniques. AI Mag 32(2):42–56
11. Leuski A, Traum DR (2010) Practical language processing for virtual humans. In: IAAI-10
12. Lukin SM, Gervits F, Hayes CJ, Moolchandani P, Leuski A, Rogers III JG, Amaro CS, Marge M, Voss CR, Traum D (2018) ScoutBot: a dialogue system for collaborative navigation. In: Proceedings of ACL
13. Marge M, Bonial C, Byrne B, Cassidy T, Evans AW, Hill SG, Voss C (2016) Applying the wizard-of-oz technique to multimodal human-robot dialogue. In: Proceedings of RO-MAN
14. Marge M, Bonial C, Pollard KA, Artstein R, Byrne B, Hill SG, Voss C, Traum D (2016) Assessing agreement in human-robot dialogue strategies: a tale of two wizards. In: International conference on intelligent virtual agents. Springer, pp 484–488
15. Marge M, Bonial C, Foots A, Hayes C, Henry C, Pollard K, Artstein R, Voss C, Traum D (2017) Exploring variation of natural human commands to a robot in a collaborative navigation task. In: Proceedings of the first workshop on language grounding for robotics, pp 58–66

16. Marge M, Bonial C, Lukin S, Hayes C, Foots A, Artstein R, Henry C, Pollard K, Gordon C, Gervits F, Leuski A, Hill S, Voss C, Traum D (2018) Balancing efficiency and coverage in human-robot dialogue collection. In: Proceedings of AI-HRI AAAI-FSS
17. Marge MR, Rudnicky AI (2011) The teamtalk corpus: route instructions in open spaces. In: Proceedings of SIGdial
18. McTear MF (1998) Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit. In: Fifth international conference on spoken language processing
19. Murphy R (2004) Human-robot interaction in rescue robotics. IEEE Trans Syst Man Cybern Part C 34(2):138–153
20. Serban IV, Sordoni A, Bengio Y, Courville AC, Pineau J (2016) Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI, vol 16, pp 3776–3784
21. Serban IV, Lowe R, Henderson P, Charlin L, Pineau J (2018) A survey of available corpora for building data-driven dialogue systems: the journal version. Dialogue Discourse 9(1):1–49
22. Traum D, Georgila K, Artstein R, Leuski A (2015) Evaluating spoken dialogue processing for time-offset interaction. In: Proceedings of SIGdial, pp 199–208
23. Traum DR, Henry C, Lukin SM, Artstein R, Pollard KA, Bonial C, Lei S, Voss CR, Marge M, Hayes C, Hill S (2018) Dialogue structure annotation for multi-floor interaction. In: Proceedings of LREC
24. Vinyals O, Le Q (2015) A neural conversational model. arXiv:1506.05869
25. Wen TH, Gasic M, Mrksic N, Su PH, Vandyke D, Young S (2015) Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In: Proceedings of EMNLP
26. Xu W, Rudnicky AI (2000) Task-based dialog management using an agenda. In: Proceedings of the 2000 ANLP/NAACL workshop on conversational systems, vol 3, pp 42–47