Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference

Machine Learning from Conversation with Humans

Awrad Mohammed Ali, Avelino J. Gonzalez
Department of Computer Science
University of Central Florida
Orlando, Florida, USA
awrad.emad@knights.ucf.edu, avelino.gonzalez@ucf.edu

Abstract

Human social learning is an effective process that has inspired many existing machine learning techniques, such as learning from observation and learning by demonstration. Hence, in this paper, we are proposing another form of social learning, Learning from a Conversation (LfC). LfC is an open-ended machine learning system in which an artificially intelligent agent learns from extended dialog with a human. Our system enables the agent to adapt to new changes based on the human input. We provide a detailed description of our system and report its performance by providing several examples that reflect our system's efficiency. Test results indicate that the prototype was successful in learning from conversation.

Introduction

Artificially intelligent conversational agents play an important role in many applications, such as flight and/or restaurant reservations (Kim et al. 2007), mobile devices, tour guidance (Aggarwal et al. 2012) and teaching (Dzikovska et al. 2011). However, many of those agents are designed for fixed tasks with fixed capabilities. Additionally, any software update requires significant effort from the programmer. As a result, researchers are investigating how to create systems that learn from interacting with their surroundings using imitation learning, a form of learning from observation that requires observing the behavior of others to perform similar actions (Wang, 1995), learning from demonstration where an expert performs a sequence of actions and provides multiple examples in front of a learner/robot (Atkeson and Schaal, 1997) and more recently learning from natural instruction (Volkova et al. 2013) where the robot learns how to perform a task by translating the user's speech into behaviors.

In this paper, we present Learning from a Conversation (LfC), which is inspired by human social learning processes. LfC works similar to humans when they learn from each other through a conversation. During a conversation, we usually learn new concepts and update existing information based on the learned concepts. Furthermore, we process new information if we have pre-existing knowledge about it by trying to either convince the speaker that there is a mistake in the spoken information or we adopt the new information if we become convinced that it is true. Following the same process, LfC can update existing knowledge and add new information.

LfC differs from other forms of machine learning by giving control to the system to decide what is correct information and what is not. This gives the machine more human-like behavior. LfC creates a more challenging task than the existing learning techniques because there are no specific actions that the agent needs to learn and there is no ultimate goal for the agent to attain.

We designed LfC to be an open-ended learner in which the user's speech is used to update the agent's information automatically and the system has the ability to accept what it thinks is new information and update existing knowledge based on the user's speech. Additionally, it can reject the new information as incorrect or challenge the human to confirm it.

In order to accomplish our objectives, LfC requires memory to keep track of existing and new incoming information, and a machine learning algorithm to determine if the provided text is something that needs to be learned or not.

This paper is organized as follows; in the next section we review related work. Later we discuss the LfC architecture. Afterward, we discuss our experiments and results. Finally, we conclude with a discussion and outline future work.

Background

There are three threads of related research relevant to our work. We first review works related to conversational systems. We then discuss existing works that apply learning from human interaction. Finally, we review conversational systems that use memory in their architectures.

Chatbots and Conversational Systems

The literature on conversational systems is very large, and a complete review is beyond the scope of this paper. Therefore, here we include a brief review of the works that influenced our current research.

Generally, conversational systems can be divided into two categories, chat-oriented systems for entertainment purposes and task-oriented systems that help the user accomplish specific tasks. As a result, researchers keep investigating ways to improve these systems to accommodate users' needs.

Hence, the existing systems have shown significant improvement compared to the earlier versions of conversational systems, but this area of research requires further efforts to improve the systems' responses to reflect an adequate understanding of user speech.

ELIZA (Weizenbaum 1966) was the earliest known conversational system. It gave hope to the AI community that the Turing test can be passed by applying several tricks. ELIZA manipulates the user input and utilizes scripts and keywords to provide its responses by using pattern-matching and by turning the user speech around with little or no contribution from the program.

Another well known chatbot was introduced by Wallace (2004) who presented the ALICE chatbot which has a large knowledge base of 40,000 categories compared to ELIZA which has only 200. ALICE uses AIML (Artificial Intelligence Markup Language), an XML language designed to provide heuristic rules for the conversation.

In 2011, IBM developed Watson, a question answering system that won a quiz show (Jeopardy) by defeating two former humans winners. For this project, IBM developed a software architecture called DeepQA that interpret questions to generate hypotheses and collect evidence for those hypotheses before providing its answer (Lally and Fodor, 2011).

More recently, M. Ali and Gonzalez (2016) presented a survey that covers some of the recent research in conversational systems. In their survey, the existing research is classified into five categories: heterogeneous systems that can assist the user to accomplish multiple tasks using the same interface; multi-model systems that use different models to generate divergent responses; systems that use memory in their architectures; systems that apply machine learning to improve the system responses; and systems designed to avoid generating meaningless responses by filtering the data, extracting relevant words to the topic of the conversation and using context in the dialogue to understand the topic.

Learning from Interaction

Learning from interacting with humans is an active area of research that has many applications. (Voyles, Morrow, and Khosla 1997) presented a robot that learns how to move and avoid obstacles from multiple demonstrations by a human. (Rybski et al. 2007) proposed interactive task training for a mobile robot where the learning process is a mixture of learning from demonstration and instructional learning. However, in (Rybski et al. 2007), the human has to specify verbally when the robot needs to start leaning. In contrast, in our LfC system, the agent can control what to learn and what should be updated in its knowledge base. This process allows natural communication with the agent without needing to worry about using specific keywords to trigger the learning process.

Later, (Mailler et al. 2009) presented the MABLE framework (Modular Architecture for Bootstrapped Learning Experiments) where the learning process involves descriptions, demonstrations and user feedback. MABLE was able to learn four different types of knowledge; definitions, rules, functions and procedures.

Learning from instruction is also applied to a mobile service robot (Meriçli et al. 2014). In their work, Meriçli et al. use a keyword-based filtering approach to search for specific words in the user's speech to execute the commands. However, these keywords need to be in the correct order. Furthermore, the system does not accept synonyms for the same words, which makes the system less practical.

Learning from natural instruction has been applied to generalizing the learning task to include similar tasks. For example, (Volkova et al. 2013) presented a procedural dialog system that learns from task-oriented textual resources using light, non-expert supervision. This system is quite flexible because it allows the users to change their goals slightly before completing the original task.

Following a similar direction, (Mohan and Laird 2014) proposed a system that learns to generalize its performance to include different variations of the task by interacting with humans using natural instructions. This work demonstrates that the learned knowledge could be transferred by applying it to different tasks that have similar structure. Both works by (Volkova et al. 2013) and by (Mohan and Laird 2014) require the agent to perform limited tasks to reach a distinct goal with some flexibility.

(Goldwasser and Roth 2011) presented an interactive robot that learns how to play a card game by using human instructions and by learning from some examples illustrating the process.

(Grizou, Lopes, and Oudeyer 2013), and (Matuszek et al. 2013) used learning from instructions to teach the agent the meaning of completely unknown instructions to perform a new task. Similarly, (Misra et al. 2014) presented a work that deals with ambiguous and incomplete instructions to perform a task by finding a valid mapping for these generalized instructions.

To accommodate a changing environment such as search and rescue, (Cantrell et al. 2012) presented a robot that learns unknown actions from natural language constructions. The new information is used to update or create new plans for the robot. Additionally, this system allows the robot to execute actions with multiple goals in parallel.

Planning under uncertainty has been also encountered by (Grizou et al. 2014), who proposed an interactive learning system that can learn a task from unlabeled instructions. This system requires the user's feedback to acknowledge whether the movement taken is correct or not. However the system has limited knowledge about what the goal should be and it does not know a priori what "correct" and "incorrect" words mean. Therefore, it needs to figure out by trial and error what the goal is and what are the right movements to attain it.

From the discussed works, clearly learning from instruction applies only to goal-oriented tasks where the agent has to learn how to accomplish a specific task with limited variations by using a specific number of instructions. In contrast, in our LfC system, the goal is to continuously learn declarative instead of procedural knowledge, with no restriction to the number of interactions with the user.

Overview of Memory Models

Learning cannot be accomplished without having some form of memory to remember previous events and utterances. Hence, using a memory is important in the learning process and also in producing coherent responses that improve the system's performance in general. Therefore, in this section we provide a brief overview of some conversational systems that use memory as part of their architectures.

The earliest and simplest form of memory was seen in the ELIZA chatbot (Weizenbaum 1966). ELIZA saves the user's previous utterances and uses them in its future responses. This is followed by many other projects that use memory as part of their architecture. As a recent example, (Laird 2008) proposed SOAR, one of the earliest artificial intelligent cognitive architectures that has a goal of creating a general computation system that has cognitive capabilities modeled after that of humans. Its architecture includes semantic memory composed of declarative knowledge and episodic memory. Additionally, (Banchs and Li 2012) introduced IRIS (Informal Response Interactive System), a chat-oriented dialogue system that learns new concepts from users and recalls previous chats with the users. Thus, the main objective of IRIS was to generate relevant responses to the user utterances. However, it has no ability to revise existing knowledge. Following the same procedure, (Kim et al. 2014) presented a spoken dialog system that uses long-term memory to save the user's previous utterances and use them later as part of the system's responses.

However, like humans, a memory system must choose to remember what might be important and forget the rest of the conversation. The same idea has been applied by (Elvir et al. 2016) as an algorithm that can extract important words from the sentence and remember them as episodic memory.

LfC Architecture

Our ultimate vision of the LfC architecture consists of several components as illustrated in Figure 1. Since the problem is very complicated and requires adding many components to make the system function as desired, the focus of this paper is on the learning process, which assumes any information from the user is reliable. As part of our future research, we will add trust prediction to the system.

It is important to mention that interaction with the LfC system is done through text rather than speech to avoid any misunderstandings that can occur because of speech recognition errors. As shown in Figure 1, the human sends a text input to the system. The system accepts the input and assesses the user's trust by checking its previous interactions with the system and by doing an online search for the level and area of expertise of that person. The trust evaluation is done only once when the user enters his/her name. Later, LfC applies text preprocessing such as, changing the user input to lower-case letters and removing punctuations which makes it easier to relate the provided utterance with the system knowledge base. Later, the system hands the processed text to the memory function, which is responsible for classifying the user text as either important information that the system might decide to learn or information that should not be learned. If the system decides that the information is important, it will continue the process; otherwise, the system discards it. The classification process is performed using Naïve Bayes classifier. After the system has determined what is important in the previous phase, the learning algorithm is used to decide if the current information is new. In this case, the system saves it in the knowledge base as new information. If the learning algorithm has encountered the given information before but it contradicts what is in the knowledge base, then the system modifies the current information in its knowledge base. Finally, if the information is already in the knowledge base, the system indicates that to the user. Hence, after each case, the system uses a dialog manager to inform the user of the status of the information and what has been updated in the knowledge base. This process is repeated until the user ends the conversation.

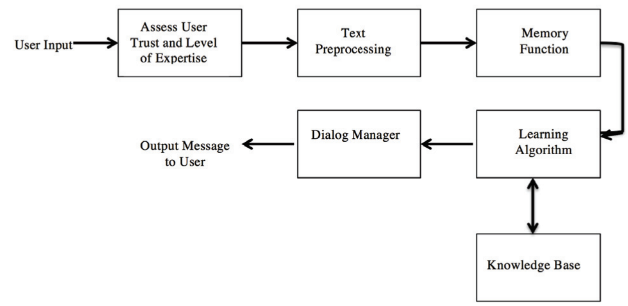

Figure 1: LfC architecture

Implementation

The core challenge in the LfC system is how to reason about and understand the given information provided by the user and determine whether there is related information in the knowledge base. Hence, our ultimate goal is to make LfC able to infer whether or not the given piece of information has an effect on existing knowledge.

The algorithm of our LfC system is shown in Algorithm 1. The system starts by training the Naïve Bayes classifier using labeled examples of sentences that are either labeled as chatting statements or not. The classifier's purpose is to indicate whether the user utterance contains important information that the system needs to consider further manipulating or only a chatting statement. Hence, our system is not designed for chatting purposes; therefore, we have limited its chatting ability to merely greeting the user and giving him/her a brief introduction of what this system is about. If the user continues chatting with the system, LfC starts introducing a topic related to the information in the knowledge base (KB) by asking the user his/her opinion about a given statement. If the user declares that the given statement is wrong, the system asks the user to correct this information. Based on the user response, the system updates its knowledge base, if necessary.

After LfC has classified the user input as important information, it uses part-of-speech tagging to analyze the tags of the user input and compares the extracted nouns with the existing nouns in each entry of the knowledge base.

Since we cannot always predict the pattern of the given sentence by the user, it is practical to use a similarity measure to decide the similarity between the text given by the user and the existing information in the system's knowledge base. Additionally, the high error rate of the available POS taggers can cause false positive matching between the user input and the system's knowledge base. Therefore, LfC uses fuzzy string matching that uses Levenshtein distance to measure the similarity between the user input and the knowledge base. The system chooses the knowledge base's entry with the highest matching score specified by the score threshold (shown in Algorithm 1) besides matching similar nouns.

Adding the fuzzy string matching improves our results significantly because it eliminates modifying irrelevant information that has partial similarity with the user input in terms of the nouns they use.

The thresholds of score matching were obtained by trial and error; hence, we found that the current thresholds give the best results using our tested knowledge base.

There are three different cases in the learning process. 1) Perfect match; when the information exactly matches that in the KB. 2) Partial match; where LfC modifies the existing information in the KB to be similar to the user input assuming the user is trustworthy. 3) No match; here LfC considers this as new information and adds it to the knowledge base assuming the human is deemed trustworthy.

In order to determine whether or not the current input is exactly similar to existing information in the KB, we used a fuzzy token-sort approach in which the strings are tokenized and sorted in alphabetical order to determine the degree of similarity. we considered sentences that achieve a similarity score of 95% and above as a perfect match. Hence, this score is chosen over 100% to make sure that small mismatches do not affect the process. The system does not update the knowledge base and only reports to the user that this information exists.

For partial matching, we compare the nouns in the user input with those of each entry in the knowledge base and to increase the confidence that the chosen entry from the KB is the right one to replace, we set up the matching score to be above 70%. We also considered the case when not all the nouns in the user input are in the KB entry; therefore, we assumed that there is a partial match if the similarity score is above 85%; otherwise, we add the information as a new entry.

Another partial matching that we encountered is when only the subject of the user input is similar to one or some entries in the KB. In this case, it also measures the similarity score and if it is $\geq$ 70%, then LfC can exchange information safely; otherwise LfC adds the information as a new entry.

In all cases, the user receives a message indicating the status of the update that occurs in the knowledge base. The process continues, until the user terminates the discussion by typing the word "bye" in his/her input.

```
while user input != termination statement do
    NaïveBayesClassifier(input);
    if input not chatting statement then
        POS(input) ;
        if input in KB or matchScore >= 95 then
            Output "similar info exist"
        else if POS(nouns) in KB and fuzz.ratio ≥ 70 or
          fuzz.ratio > 85 then
            Exchange information in KB;
            Output a statement reflecting the change
              occurred
        end
        else if NN-SUBJ in KB and fuzz.ratio ≥ 70 then
            Exchange information in KB ;
            Output a statement reflecting the change
              occurred
        end
        else
            Add new information to KB;
        end
    else
        Chat with the user;
        Introduce a topic related to the KB information;
    end
end
```

Algorithm 1: Updating Existing Knowledge

Experiments and Results

To evaluate LfC's performance, we provided several examples of how the system works. Later, we provide an experimental study of creating 100 sample entries and evaluating the system's responses to them compared to our expectations. We also highlight the factors that affect the system performance, and finally we mention the drawbacks of our system and ways to improve them in our future research.

To focus on how the system expands its knowledge base, we used WORDij (Danowski, 2013) to create a semantic network that visualizes our dataset before and after interacting with the user. In Figure 2, LfC had only twelve facts related to the topic of food. Hence, in Figure 3, after 30 interactions with the user, the system's knowledge base has increased significantly.

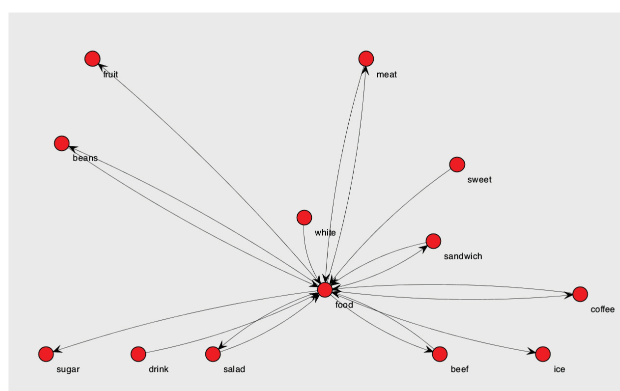

Figure 2: Semantic network that represents the knowledge base before interacting with the user

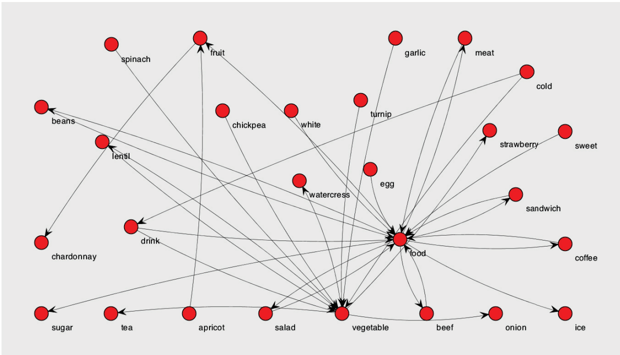

Figure 3: Semantic network that represents the knowledge base after 30 interactions with the user.

Table 1 shows some examples of the system's responses regarding the three phases of information update: perfect matching, updating existing information, and adding new information respectively. In this table, we report the similarity score between the user input and the chosen entry from the knowledge base.

Table 1: Information handling Example

Speaker	Turn	Score
User LfC	Are coffee beans fruit pits? Yes, I have the exact information in my KB coffee beans are fruit pits	100%
User LfC	banana has 90 calories You have changed banana has 105 calories by banana has 90 calories	91%
User LfC	Apple is rich in dietary fiber LfC added this information to its KB	below 70%

Table 2 shows an example of how the program starts and how it responds to chatting statements. As shown, in turn 5, the user tried to continue chatting with the system; therefore, the system decided to change the subject and asked the user a question from its knowledge base to force him/her to talk about relevant information to its knowledge base.

It is important to mention that there is a tradeoff between adding new information and modifying existing information, depending on the matching threshold. However, we think that modifying the wrong information is worse than adding new information that could have been used to modify existing knowledge. For example, by decreasing the matching score from 70% to 20%, the system considers "apples are rich in vitamins and minerals" and "apples are rich in dietary fibers" to be partially similar because it matches "apples" and "rich" from both sentences. However, they in fact contain different information. Therefore, the system should not modify its existing knowledge and add the provided statement as a new entry to the knowledge base.

As we stated earlier, to evaluate LfC's performance we created 100 entries that varied between chatting statements and statements that are related to the system's knowledge base. For this task, we used a knowledge base that contained

Table 2: Chatting Example

No.	Speaker	Turn
1	LfC	Hi, enter your name please.
2	User	Awrad
3	LfC	hi Awrad! how are you today?
3	User	doing alright
4	LfC	Amazing, What is your input?
5	User	Hi again
6	LfC	Hi, I hope you are doing well. I am an AI agent that needs your help to verify and update my information related to food. Do you think that strawberries have more vitamin C than the oranges?
7	User	I don't think so
8	LfC	Then, could you correct my information please?

3000 statements of general facts. Before testing our system using the 100 entries, we estimated the system's responses related to each entry, i.e., ignore it when it is a chatting statement; confirm it when it already exists in the knowledge base; use it to update an existing knowledge base entry or add it as a new entry.

We marked how many times the system matched our expectation and how many times it did not. Out of the 100 queries, the system had 12 misses, in which four of those misses were chatting statements that the Naïve classifier classified as relevant information; therefore, our LfC system added them to its knowledge base. In the rest of the misses, LfC added information when our expectations were that it should have replaced existing knowledge. Hence, the second case was expected because we set up the matching score to be ≥ 70; therefore, even when there is only a partial matching between the knowledge base and the user entry, the matching score can be lower than this threshold. As mentioned earlier, these values were chosen to eliminate any risk of overwriting correct information in the knowledge base with incorrect information.

Conclusion

In this paper, we presented LfC, a learning from conversation system that updates and modifies its existing information based on the user speech. The system differs from existing approaches by its ability to lead the learning process by discarding information when it thinks it is unimportant and modifying or adding knowledge when LfC believes the user has better knowledge about it. There are several factors that can affect LfC's performance, including spelling mistakes in the user input, misclassifying the user input (i.e., chatting statement or not), and incorrect speech tagging. Therefore, we are planning to upgrade our system to include a spell checker and use multiple classification algorithms and perform a majority vote to determine the classification of the user input.

For our future research, we are planning to add trust prediction based on the user's level of expertise. This will help prevent including false information to the system's knowledge base from non-trustable users. Also, we plan to improve the system to be more robust to factors that can affect its understanding of the user utterance. We are also planning to upgrade the current system to include entity linking that is used to match different words with similar meaning. We are also planning to evaluate LfC in a larger, more realistic settings, such as using crowdsourcing. Moreover, we are considering using more NLP tools, such as named entity recognition and dependency parser.

References

Aggarwal, P.; Artstein, R.; Gerten, J.; Katsamanis, A.; Narayanan, S.; Nazarian, A.; and Traum, D. R. 2012. The twins corpus of museum visitor questions. In *The International Conference on Language Resources and Evaluation (LREC)*, 2355–2361.

Atkeson, C. G., and Schaal, S. 1997. Robot learning from demonstration. In *Proceedings of International Conference on Machine Learning*, volume 97, 12–20.

Banchs, R. E., and Li, H. 2012. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the Association for Computational Linguistics (ACL), System Demonstrations*, 37–42.

Cantrell, R.; Benton, J.; Talamadupula, K.; Kambhampati, S.; Schermerhorn, P.; and Scheutz, M. 2012. Tell me when and why to do it! run-time planner model updates via natural language instruction. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 471–478. IEEE.

Danowski, J. A. 2013. Semantic network analysis software. University of Illinois at Chicago.

Dzikovska, M. O.; Isard, A.; Bell, P.; Moore, J. D.; Steinhauser, N. B.; Campbell, G. E.; Taylor, L. S.; Caine, S.; and Scott, C. 2011. Adaptive intelligent tutorial dialogue in the beetle ii system. In *International Conference on Artificial Intelligence in Education*, 621–621.

Elvir, M.; Gonzalez, A. J.; Walls, C.; and Wilder, B. 2016. Remembering a conversation–a conversational memory architecture for embodied conversational agents. *Journal of Intelligent Systems*.

Goldwasser, D., and Roth, D. 2011. Learning from natural instructions. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Grizou, J.; Iturrate, I.; Montesano, L.; Oudeyer, P.-Y.; and Lopes, M. 2014. Interactive learning from unlabeled instructions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*.

Grizou, J.; Lopes, M.; and Oudeyer, P.-Y. 2013. Robot learning simultaneously a task and how to interpret human instructions. In *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, 1–8. IEEE.

Kim, S.; Lee, C.; Jung, S.; and Lee, G. G. 2007. A spoken dialogue system for electronic program guide information access. In *Robot and Human interactive Communication*, 178–181. IEEE.

Kim, Y.; Bang, J.; Choi, J.; Ryu, S.; Koo, S.; and Lee, G. G. 2014. Acquisition and use of long-term memory for personalized dialog systems. In *Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction*. 78–87.

Laird, J. E. 2008. Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications* 171:224.

Lally, A., and Fodor, P. 2011. Natural language processing with prolog in the ibm watson system. *The Association for Logic Programming (ALP) Newsletter*.

M. Ali, A., and Gonzalez, A. J. 2016. Toward designing a realistic conversational system: A survey. In *Proceedings of the 29th Flairs conference*, 2–7.

Mailler, R.; Bryce, D.; Shen, J.; and O'Reilly, C. 2009. Mable: A framework for learning from natural instruction. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 393–400.

Matuszek, C.; Herbst, E.; Zettlemoyer, L.; and Fox, D. 2013. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, 403–415.

Meriçli, C.; Klee, S. D.; Paparian, J.; and Veloso, M. 2014. An interactive approach for situated task specification through verbal instructions. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 1069–1076.

Misra, D. K.; Sung, J.; Lee, K.; and Saxena, A. 2014. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. In *Proceedings of Robotics: Science and Systems (RSS)*.

Mohan, S., and Laird, J. E. 2014. Learning goal-oriented hierarchical tasks from situated interactive instruction. In *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*, 387–394.

Rybski, P. E.; Yoon, K.; Stolarz, J.; and Veloso, M. M. 2007. Interactive robot task training through dialog and demonstration. In *International Conference on Human-Robot Interaction (HRI)*, 49–56. IEEE.

Volkova, S.; Choudhury, P.; Quirk, C.; Dolan, B.; and Zettlemoyer, L. S. 2013. Lightly supervised learning of procedural dialog systems. In *Proceedings of the Association for Computational Linguistics (ACL)*, 1669–1679. Citeseer.

Voyles, R. M.; Morrow, J. D.; and Khosla, P. K. 1997. Towards gesture-based programming: Shape from motion primordial learning of sensorimotor primitives. In *Robotics and Autonomous Systems*, volume 22, 361–375. Elsevier.

Wallace, R. S. 2009. *The anatomy of ALICE*. Springer.

Wang, X. 1995. Learning by observation and practice: An incremental approach for planning operator acquisition. In *Proceedings of International Conference on Machine Learning*, 549–557.

Weizenbaum, J. 1966. Eliza a computer program for the study of natural language communication between man and machine. In *Communications of the ACM*, volume 9, 36–45. ACM.