Conversational Learning

Forough Arabshahi
Carnegie Mellon University
farabsha@cs.cmu.edu

Kathryn Mazaitis
Carnegie Mellon University
krivard@cs.cmu.edu

Toby Jia-Jun Li
Carnegie Mellon University
tobyli@cs.cmu.edu

Brad A. Myers
Carnegie Mellon University
bam@cs.cmu.edu

Tom Mitchell
Carnegie Mellon University
tom.mitchell@cs.cmu.edu

Abstract

Although machine learning has been highly successful in recent years, this success has been based on algorithms that exhibit only one of the multiple learning paradigms used by humans: learning statistically from many examples. Here we consider a second learning paradigm widely exhibited by humans, but rarely by computers: learning from instruction involving natural language conversations and demonstrations. We argue that this second paradigm – conversational machine learning – is ripe for rapid research progress, and that it holds the potential to make it possible for every user of a computer or mobile device to become a programmer. We define the problem of conversational learning, survey relevant literature, and provide as a case study the Learning from Instruction Agent (LIA) project. Finally we lay out a set of future research directions involving grounded conversational instruction that appear to be key to progress in this area.

1 Introduction

Language affects our perception of our world, and language and thought are interconnected in fundamental ways (Chomsky, 2017; Boroditsky, 2011; Vygotsky, 1964). We humans teach one another about new concepts through natural language conversations, which helps us learn from past experiences of others.

The advent of conversational agents in the past few years has made it possible for humans to interact in a new way with their computers: through natural language conversations. Despite the progress made so far (Wen et al., 2017; Young et al., 2013; Bordes et al., 2017; Eric and Manning, 2017; Ghazvininejad et al., 2018; Liu et al., 2018a), we are still in the early stages of understanding how to take advantage of this new conversational capability, and current uses of conversational systems remain limited to invoking simple pre-programmed capabilities such as booking travel reservations, checking the weather and setting alarms.

Conversations between humans are far more rich than current conversations between humans and machines. Human to human conversation is used to debate issues, to plan joint activities, to teach one another, and more. In this paper, we consider the use of conversational systems to enable humans to instruct their computers interactively, effectively to program their computers by natural language instruction, much as they would instruct another human. For example, a user might wish to instruct their phone that "Whenever it snows at night, set my alarm 30 minutes earlier." If the phone does not currently understand how to implement this instruction, the user might then break it into step-by-step instructions, "To find out whether it is snowing, open the weather app and look here at current conditions." "To set my alarm 30 minutes earlier, open the alarm app and subtract 30 minutes from the wake-up time." Note here the form of instruction may involve interleaved or simultaneous conversation and demonstration.

This kind of conversational instruction, if it can be made successful, would have a dramatic impact on the relationship that humans have with their computers. Today, less than 1% of humans have the ability to program new capabilities into their phones or computers – instead, users view their phones as having a fixed set of capabilities that have been pre-programmed at the factory. If it was possible for each user to extend the functionality of their phone or computer through natural language conversation, then we would suddenly find ourselves in a world where 99% of users have the ability to program and customize their computers to their specific needs. As computer-human teaming becomes increasingly prevalent in the workplace, this ability of workers to modify the function of their computer teammates might have an important impact on how AI influences future jobs – in particular whether AI replaces or augments human workers.

A second perspective on conversational learning is that it can become an important new paradigm to complement the current big-data statistical paradigm that currently dominates the field of machine learning. Statistical learning, while important, is just one of the learning paradigms used by humans. If conversational instruction can be made successful, then it is likely to become a fundamental tool for machine learning, and we are likely to see a growing volume of research that blends machine learning from big data statistical analysis with interactive conversations with human experts.

This research is related to what others have called "end-user programming" – EUP (Ko et al., 2011)), and there are already a number of initial systems that exhibit some of the desired properties (e.g., (Goldwasser and Roth, 2014; Azaria et al., 2016; Wang et al., 2016; Srivastava et al., 2017b; Laird et al., 2017; Labutov et al., 2018).).

The goal of the current paper is to explore today's state of the art in research relevant to conversational learning. We first review a variety of relevant research, then consider in more detail a specific case study (the Learning by Instruction Agent, LIA), discussing lessons learned and future research directions.

2 Literature Review

The idea of learning from an expert teacher has been studied in many different contexts, from reinforcement learning (Luketina et al., 2019), to imitation learning (Liu et al., 2018a; Wang et al., 2019), to learning from a teacher agent who provides a minimal set of labeled examples that uniquely specifies the target concept (Dasgupta et al., 2019). In much of this work, communication between the machine and teacher is in some formal language, or the machine passively observes the expert teacher. Although we will briefly describe these efforts, our focus in this paper is on a more under-studied scenario: teaching through natural language instruction, demonstration, and interactive conversation. In this case, the machine deliberately attempts to learn a new task and a teacher explicitly teaches it to the machine. This explicit teaching scenario can be categorized as an instance of L2 learning as defined in (Laird and Mohan, 2018), who point out the distinction in human learning between learning mechanisms that are innate and subconscious such as synaptic plasticity (L1 learning), versus learning mechanisms that are deliberately invoked and may involve explicit planning and reasoning.

2.1 Instructing Novel Tasks

In this sub-section, we review the current state of the art in conversational learning, where the machine learns a novel task through conversation. This is also referred to as interactive task learning (Laird et al., 2017). Permitting the end user to specify new tasks in terms of the machine's current capabilities is what separates the systems below from utilities like IFTTT (`https://ifttt.com/`) and Zapier (`https://zapier.com/`).

Intelligent agents and robots One of the intelligent agents that uses natural language instructions is the Learning by Instruction Agent (LIA) (Azaria et al., 2016; Srivastava et al., 2017a,b; Labutov et al., 2018; Lu et al., 2019). LIA is an intelligent assistant that runs on a user's smart-phone and can interact with certain phone applications. End-users can teach LIA to do new tasks in the domain of the supported phone applications through natural language instructions.

Another interactive agent that can be instructed in natural language is Rosie (Mohan et al., 2012). Rosie was developed in the SOAR architecture (Laird et al., 1987) and uses natural language instructions to learn how to perform novel goal-oriented tasks in the context of manipulating real world objects using a robot arm (Mohan and Laird, 2014; Mininger and Laird, 2018). Similar to LIA, Rosie's learning is incremental. Rosie relies on a broad range of information including perceptual, semantic, and procedural knowledge for learning. Moreover, the situated interaction mechanism in Rosie is an effective way of learning grounded representation of words in the environment.

Both Rosie and LIA control their learning by requesting instruction about unknown concepts. This mixed initiative interaction results in efficient learning since the instructor can rely on the agent to initiate an interaction if needed. Additionally, Rosie and LIA dynamically extend their language capabilities and their understanding of their environment through interaction.

Teaching Games with Natural Language Because games are well-structured, they provide a medium for clearly defining the problem space and task and for explicitly grounding concepts. This makes them good candidates for conversational learning methods. Goldwasser and Roth (2014) proposed learning from natural language instructions and framed it as a problem of mapping natural language statements to a semantic interpreter that extracts the meaning of the stated action. As is the case with conversational learning, here the learner does not have access to examples labeled with the correct semantic interpretation and only relies on the feedback acquired through interaction based on the predicted interpretation. Their proposed algorithmic learning approach iteratively performs the following steps: generate rules from natural language instructions, receive feedback by acting in the world, and update the semantic interpreter accordingly. They test the proposed learning by instruction framework by playing the Freecell solitaire card game.

Wang et al. (2016) proposed a language learning setting in which the computer learns to map utterances to semantic interpretations through playing a block game called SHRDLURN with a human player. The underlying learning paradigm is in essence similar to (Goldwasser and Roth, 2014), but (Wang et al., 2016) showed that in order to improve the efficiency of language learning, the computer should be able to explicitly reason about the human and introduced a pragmatics model to do this reasoning.

Another agent that uses restricted natural language interactions for learning new games is $\text{Rosie}_{\textbf{TAG}}$ (Kirk and Laird, 2014). This agent first learns the formulation of the game (i.e. the objects, players, game rules etc.) through interactive instructions and then uses its own general strategies to solve the game. In follow-up work, this agent was further extended to interactively learn in a one-shot setting where the agent relies on learning hierarchical symbolic representations of task knowledge (Kirk and Laird, 2019).

Instructive Demonstrations Demonstration is another common modality of task instructions. The use of the programming by demonstration technique (McDaniel and Myers, 1998; Myers, 1987; Cypher and Halbert, 1993; Lieberman, 2001) for users to teach intelligent agents has been explored in prior literature (e.g., (Leshed et al., 2008; Intharah et al., 2019; Li and Riva, 2018; Chao et al., 2011; Liu et al., 2018b)). Naturally, humans teach each other using a combination of conversational instructions and demonstrations (Li et al., 2018a), which we refer to as teaching by showing and telling. Combining these two modalities can support more natural end user programming experiences (Myers et al., 2017). This multi-modal approach dates back to early systems like Put-that-there (Bolt, 1980), where direct manipulation inputs were used to clarify spatial references used in natural language instructions. Later work such as PLOW (Allen et al., 2007), Rosie (Mohan et al., 2012), and APPINITE (Li et al., 2018a) further explored this multi-modal approach, introducing new interactive techniques for grounding, disambiguating, and clarifying natural language instructions using demonstrations. However, there are still many remaining challenges preventing wide adoption of this approach, such as generalizing learned tasks, handling new task domains, supporting greater user expressiveness, and enabling more natural user interactions. We believe that an interesting research direction can be to develop new models that use both language instructions and demonstrations for teaching novel tasks to computers.

Programming with Natural Language Shifting the nature of the language used to represent computer programs from one optimized for machines to something that looks and behaves more like natural language has been a goal since at least the development of COBOL in 1957, and arguments about natural language programming were already heated in 1966 (Halpern, 1966). More recently there have been efforts to create coding platforms with "natural language-like" syntax for writing interactive fiction (Reed, 2010) aimed at authors who are not programmers. Price et al. (2000) proposed a natural language-based user interface for creating, modifying and examining Java programs, and Le et al. (2013) proposed a programming system for synthesizing smartphone automation scripts from natural language descriptions. These works mostly attempt to directly map natural language utterances to programs, and otherwise behave like traditional programming languages in their treatment of syntax and logic errors. Good and Howland (2017) found that for novices, a programming language that appears similar to natural language, but nevertheless has all the rigid requirements of traditional programming, can cause more harm than good.

To address this shortcoming, researchers have been developing interactive natural language programming paradigms. Wang et al. (2017) tried to create a natural language interface for users to manipulate blocks in a simulated game. Their goal was to "naturalize" a programming language through conversational interactions such that the resulting programming language is more natural language-like than programming language-like. Fast et al. (2018) proposed a conversational agent that can perform open-ended data science tasks through natural language conversation with users. These works and other learning by instruction and demonstration works such as (Goldwasser and Roth, 2014; Wang et al., 2016; Azaria et al., 2016; Li et al., 2018a, 2019) lay the foundation for making natural language interfaces for complex tasks such as analyzing data, manipulating texts, and querying databases and knowledge bases, or simple tasks covering everyday needs such as setting alarms conditionally and sorting email.

2.2 Natural Language Interactions

Natural language interactions have been used in various applications such as chatbots, goal oriented conversational agents, navigation, reinforcement learning, imitation learning and others. These applications tend not use natural language to teach how to accomplish novel tasks, but rather to provide a more natural interface for users to communicate with the computer system.

Instruction in Reinforcement Learning Natural language instructions have appeared in the Reinforcement Learning (RL) community to improve the generalization performance and the sample efficiency of RL. Luketina et al. (2019) provides a comprehensive list of recent work. We review here some of the models that use natural language instructions to teach new tasks to agents in a RL context. Co-Reyes et al. (2019) proposed an interactive setting where a series of natural language corrections guides the agent to acquire a desired new task. They show that natural language corrections are substantially more informative than simpler forms of supervision, such as preferences, while being substantially easier and more natural to provide than reward functions or demonstrations. Branavan et al. (2009, 2010) proposed a RL-based framework that converts a sequence of actions given in natural language (e.g. instructions for how to delete a file) to actions. In this work, each instruction maps to a sequence of commands which are not necessarily stated in the original instruction, and the challenge is to infer this hidden information. They achieve this by developing statistical models that rank the generated candidate actions. Misra et al. (2017) proposed a RL-based method that jointly maps visual and natural language instructions to actions. This model separately induces text and environment representations and combines them to output a policy. While this representation is good for capturing coarse correspondence between different modalities, it does not encode lower-level mappings between specific positions on a map. Therefore, Janner et al. (2018) proposes to combine the language and environment representations in a spatially localized manner to improve performance.

RL approaches are often highly sample inefficient. Reward shaping is an approach for reducing an RL agent's interaction time with the environment by carefully designing reward functions. Although effective, designing appropriate shaping rewards is difficult as well as time-consuming. Goyal et al. (2019) proposed using instructions for reward shaping in RL to overcome this challenge and showed that language-based rewards lead to successful completion of the task 60% more often on average, compared to learning without language.

Navigation Natural language instructions have also been used for navigation in an instruction following manner (MacMahon et al., 2006; Fried et al., 2018a; de Vries et al., 2018; Chevalier-Boisvert et al., 2018). These instructions typically present strategies for solving a certain task (e.g. directions for finding an object) and are not directly categorized as conversational learning from our perspective. Some of these works purely use natural language instructions (MacMahon et al., 2006; Tellex et al., 2011; Fried et al., 2018a) while others use a blend of natural language instructions and vision for navigation (Fried et al., 2018b; Chen and Mooney, 2011; Kim and Mooney, 2013; Anderson et al., 2018; Wang et al., 2019). For example, Chen et al. (2019) introduced a task and dataset for navigation and spatial reasoning called touchdown, in which the goal is to use natural language instructions to navigate to a certain location in a real-life urban environment and find a hidden object.

Instruction in other contexts Natural language instructions have also been used in decision making (Hu et al., 2019), semantic parsing (Artzi and Zettlemoyer, 2013), playing games (Reckman et al., 2010), building knowledge graphs (Hixon et al.,

2015) and human-robot interactions (Bisk et al., 2016). In most of these works, instructions provide a description for solving tasks rather than teaching the problem specification.

3 Learning from Instruction and Programming By Demonstration Agents

The Learning from Instruction Agent (LIA) provides one case study of conversational learning. LIA is a prototype intelligent digital assistant that runs on a user's Android smart-phone, and can be instructed in natural language. For example, the user can teach LIA "whenever a meeting is added to my calendar on a Saturday, tell my spouse." If LIA does not understand how to perform this task she will ask the user to teach it by breaking the task down into a sequence of steps that LIA understands. For the case of this example, the user can say, "(first) if the Weekday field of a new meeting is Saturday, then first create a new text message, (second) in the Recipient field of the new text message, put my spouse's phone number. (third) in the body of the text message say I just agreed to a Saturday meeting, (fourth) send the message" After this teaching session, LIA can perform this task, and will attempt to generalize the procedure to cover similar future commands.

A typical conversation held with the current LIA system is shown in Figure 1. As described in the figure caption, the user here teaches LIA a new procedure, resulting in LIA updating its semantic parser so that future parses of this (and similar) sentences will construct a logical form which, when executed, performs the intended procedure.

LIA begins with a library of sensors and effectors that permit interaction with the world around it. For example, the phone's GPS is a sensor; creating and activating a timer on the phone is an effector; the user's email account can take on both roles depending on whether you're reading mail (sensor) or sending it (effector). LIA also has an internal long term memory whose read and write operations are treated as an additional sensor and effector, respectively. LIA starts from a limited grammar that allows it to semantically parse simple commands to access each sensor and effector (e.g., "set a timer," "check for new email"). Over time it extends this semantic grammar as it interacts with the user and adds new functionalities. In addition, LIA's parser is continually trained from user feedback, allowing it to adapt to the user's phrasing habits over time. Therefore LIA's learning task is 3-fold:

1. Learn new symbolic knowledge by adding new entities and facts to its internal memory, or knowledge graph.

2. Learn new procedural knowledge by associating a new natural language phrase with a sequence of previously known steps.

3. Learn a refined strategy for semantic parsing, through online training from candidate parses ultimately accepted by the user.

LIA uses a grammar-based semantic parser built on SEMPRE (Berant et al., 2013), where each grammar rule performs both a syntactic operation, and a semantic operation of composing the logical forms of its constituents into a single new logical form. LIA's world knowledge can be built by the user through natural language instruction. For example, users can create new concepts, such as the concept of a "colleague" by an instruction such as "A colleague is a kind of person." Users can add fields to a concept, for example, "Most colleagues have a work phone number." They can also create instances of that concept, e.g. "Sally is a colleague and her phone number is 555-1234." This allows the user to describe the world in an object-oriented programming style through natural language. In the back-end, this information is stored in LIA in a knowledge graph built on top of Theo (Mitchell et al., 1991) which represents frame-style knowledge in terms of `(entity, relation, value)` triples, in which any triple can also be treated as an `entity`. Theo's uniform data structure naturally supports a generalization hierarchy of arbitrary depth, and allows representing beliefs about beliefs. For example, if we wanted, LIA could represent that we learned about Sally's phone number in a particular conversation, that the conversation took place on a particular date, and that her phone number changed on a particular date.

3.1 Incorporating Demonstrations: Show and Tell Instruction

In human to human instruction, natural language conversation is sometimes combined with demonstration. There have been attempts to create instructable phone agents that similarly combine language with demonstrations to provide a "show and tell" style interface (Li et al., 2017, 2018a, 2019).

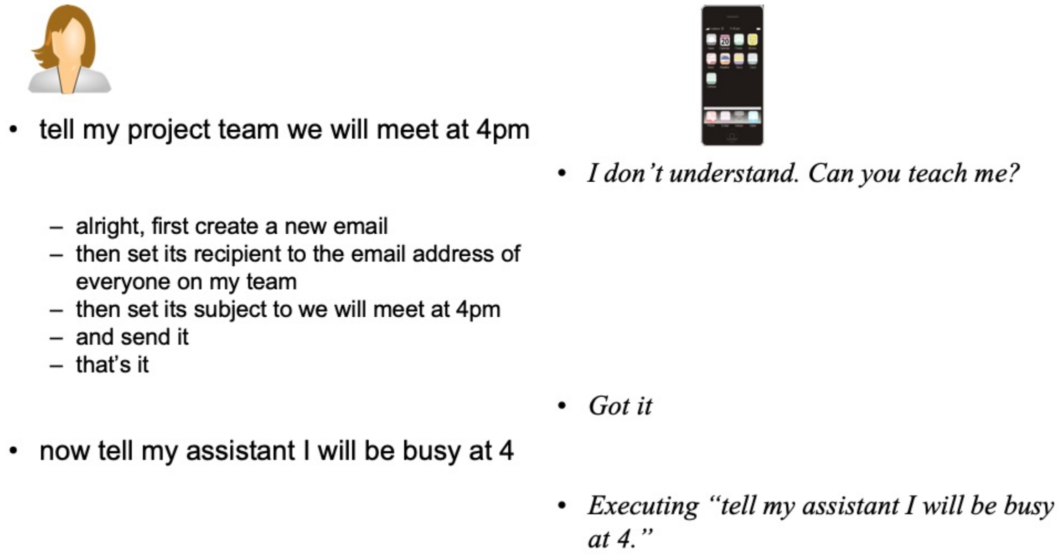

Figure 1: **A sample instructional conversation with LIA.** User (left) gives a command which LIA (right) fails to understand, so LIA invites the user to teach it. Once the user instructs LIA how to perform the command, LIA updates its semantic grammar and parse strategy so that parsing the original "tell my project team we will meet at 4pm" produces a logical form which, when evaluated, performs that full command. As shown in the final step of the conversation, when the user makes a subsequent request to "tell my assistant I will be busy at 4," LIA performs the command without guidance because it has successfully generalized from the previously given instructions.

For example, to teach one's phone to "Set my alarm 30 minutes earlier whenever it snows at night," it would be easier to demonstrate how to check current conditions on the weather app, than to use language to describe how. Demonstrations play three main roles in such a combined "show and tell" system: (1) disambiguating unclear instructions; (2) grounding user instructions involving arbitrary phone apps; and (3) providing contexts for inferring parameterizations in the instructions.

GUI demonstrations help users focus on the key differentiating factors between possible actions resulting from ambiguous instructions (Li et al., 2018a). For example, suppose a user instruction in a restaurant-reservation task is, "select a steakhouse in downtown." It is unclear what to do when there are multiple steakhouses in downtown. The study in (Li et al., 2018a) found that by having users demonstrate the action on the GUI of a relevant app and then verbally explain the criteria for choosing among the confusing items while viewing these items highlighted on the GUI, one can effectively direct users to focus on the differentiating factors between these items. Additionally, this activity guides users to use the vocabulary and structures visible on the app GUI, making natural language understanding much easier.

Users can ground unknown procedures and concepts using demonstrations on existing app GUIs. Because a demonstration system can track user activity through the accessibility interface, the effective domain can encompass the entire Android platform. By contrast, the natural-language-only instruction systems such as (Azaria et al., 2016) implement their lowest-level tasks on a by-topic basis (email, weather, etc). If a user mentions an out-of-domain task *procedure* (e.g., register for a course), the instructable agent will not be able to handle it even if the user can verbally break down the procedure (e.g., "first, search for all available courses…") due to the lack of support for the task domain (Li et al., 2018b). The programming by demonstration technique allows users to teach such procedures by demonstrating using existing mobile apps (Li et al., 2017). Similarly, users can employ demonstration to ground unknown *concepts* used in their natural language instructions (Li et al., 2019). For example, if a user verbally explains the concept "a good course" as "a course with a high course rating" when the system does not understand what the "course rating" is, the user can demonstrate finding out the course rating in an app GUI.

When both the user's natural language instruction and the corresponding demonstration are available, one can leverage the demonstration and the hierarchical structures in the underlying app GUIs to infer both the parameters of the generalized instruction and their legal values (Li et al., 2017; Li and Riva, 2018). For example, when a user demonstrates how to do the task "order a cup of iced cappuccino", the system can associate "iced cappuccino" in the user utterance with the demonstrated action of choosing the item "iced cappuccino" in a menu on the Starbucks GUI screen, infer it as a task parameter, and extract all the other beverage options in the GUI menu as alternative values for this parameter.

4 Lessons Learned and Future Directions

Here we summarize lessons from recent relevant research and suggest future directions.

4.1 User Tests

A user study on an early version of LIA, run with Mechanical Turkers, showed that people were able to instruct LIA on a number of email-related tasks, then re-use the functions they taught to reduce their effort by 39% (Azaria et al., 2016). The Mechanical Turk study of Wang et al. (2017), where they attempt to "naturalize" a programming language through interaction as explained in Section 2, showed that around 85% of the people tended to use the naturalized language vs. the core programming language. Moreover, their study showed that different people have different preferences for the way they define new functionalities.

LIA's user study also showed that only half of the people were able to finish all the given tasks. Also, in (Wang et al., 2016) 20% of the users did not play the game using natural language and instead tried to solve the puzzles by scrolling. There is still much room for improving these systems.

4.2 Need for Common Sense

One of the most important lessons from LIA is that users often give instructions that may be considered reasonable if given to a human, but which nevertheless underspecify their intent. For example, a user teaching LIA might give the instruction "whenever it snows at night, wake me up 30 minutes earlier". If the user's intention was to account for traffic slowdowns caused by snow in their commute to work, then what they probably meant was "whenever it snows enough to cause traffic slowdowns and it's a work day, wake me up 30 minutes earlier". In order to create instructable agents that match the teaching style of most users, our models should attempt to uncover these under-specified preconditions. In this sense, learning from human instruction may require substantial common sense reasoning, and the ability to have clarification dialogs with the user.

Note in the above example, if the user had stated "whenever it snows at night, wake me up 30 minutes earlier *because I want to get to work on time.*", then the computer would have more information to reason about how and whether the requested action would achieve the intended goal. This has been referred to as a purpose clause as early as the 90s (Di Eugenio, 1992) and a benchmark dataset and commonsense reasoning framework was recently proposed for it in Arabshahi et al. (2020). The purpose clause has also been considered in goal-oriented dialog agents for improving task interpretation (Mohan and Laird, 2014; Mininger and Laird, 2018).

4.3 Generalizing from Specific Scenarios

Another challenge involves generalizing instruction beyond the specific example used for teaching. For example, if the user teaches "when it snows at night wake me up 30 minutes earlier", the agent should be able to then handle "when the weather forecasts rain, set my alarm for 1 hour later" without needing to be taught how to do this. Lu et al. (2019) attempted to tackle this problem by developing a semantic parser capable of one-shot semantic parsing, though performance was far from ideal. Advancement in few-shot semantic parsing remains to be further explored (Ferreira et al., 2015; Bapna et al., 2017; Herzig and Berant, 2018; Dadashkarimi et al., 2018).

We humans rely on mixed initiative clarification dialogs to handle vague or misunderstood statements. Agents that learn by instruction should be equipped with this same ability. Hixon et al. (2015) showed that deploying a mixed initiative strategy doubled the knowledge acquisition rate. When knowledge about the current task is missing, it may be acquired by asking the human, under the condition that the agent can correctly assess what knowledge is missing, and that it can formulate a "descriptive question" accordingly. Both of these conditions are interesting unsolved challenges.

4.4 Modifications over Time

Learning agents require the ability to update their knowledge over time because (1) they may acquire incorrect knowledge, (2) changes in the world may require corresponding changes in the agent, and (3) users might change their mind over time about how the agent should behave. In order for these updates or corrections to occur, the agent should first be able to self-inspect and explain its current world knowledge and taught procedures, so the user can detect what must be changed. This is challenging because it can require new capabilities to allow the user to edit earlier taught procedures with the same flexibility one would use if teaching a human. For example, if the user states "I said to send email when telling my mother something, but you should use text messaging to tell my brother.", the agent may have to inspect and refine multiple procedures, and store this advice in a way that will become a default during future instruction.

4.5 Combining Showing and Telling

Humans rely on a combination of demonstration and conversation to teach one another. Allen et al. (2007)'s PLOW system, as well as systems such as (Li et al., 2017, 2018a, 2019), explored many different aspects of how demonstrations can be used in conjunction with natural language instructions in multi-modal interfaces to support a more flexible, expressive, robust, and generalized task learning process.

One lesson learned is that the existing app GUIs can be great resources for task instruction. They encapsulate rich knowledge about the flows of the underlying tasks and the properties and relations of relevant entities in a structured way. Users tend to be familiar with app GUI's, making them ideal mediums that users can refer to during instruction. According to (Li et al., 2019), a major challenge in natural language instruction is that users do not know what concepts or knowledge the agent already knows so that they can use it in their instructions. Therefore, they often introduce additional unknown concepts that are either unnecessary or entirely beyond the capability of the agent (e.g., explaining "hot" as "when I'm sweating" when teaching the agent to "open the window when it is hot"). By using the app GUIs as a medium, one can effectively constrain the users to refer to things that can be found out from some app GUIs (e.g., "hot" can mean "the temperature is high"), which mostly overlaps with the "capability ceiling" of a smartphone-based agent, and allow the users to define unknown concepts by referring to app GUIs (Li et al., 2017, 2019).

An interesting future direction is to better extract semantics from app GUIs so that the user can focus on high-level task specifications and personal preferences without dealing with low-level mundane details (e.g., "buy 2 burgers" means setting the value of the textbox below the text "quantity" and next to the text "Burger" to be "2"). Some works have made early progress in this domain (Liu et al., 2018c; Deka et al., 2016) thanks to the availability of large datasets of GUIs (e.g., (Deka et al., 2017)). Recent reinforcement learning-based approaches and semantic parsing techniques have also shown promising results in learning to navigate through GUIs for user-specified task objectives (Liu et al., 2018b; Pasupat et al., 2018). For interactive task learning, an interesting future challenge is how to combine these user-independent domain-agnostic machine-learned models with the user's personalized instructions for a specific task. This will likely require a new approach of mixed-initiative instruction (Horvitz, 1999) where the agent can be more proactive in guiding the user and take more initiative in the dialog. This could be supported by improved background knowledge and task models, and a more flexible dialog framework that can handle the continuous refinement and uncertainty inherent to natural language interaction, as well as the variation in user goals likely to occur as a result of the agents involvement.

5 Conclusions

Conversational learning – machine learning through interactive dialog and demonstration with human instructors – holds the potential to change fundamentally the relationship between computer users and their computers. Whereas today most computer users have no way to program their machines, if successful this line of research could give every user the ability to program without learning a programming language, as the computer instead learns the natural instructional language of the user. While the advent of usable speech interfaces over the past decade has led users to become comfortable conversing with their computers, the conversations we have to date remain primitive. Our research challenge is to build conversational systems that support the richness of communication that conversation provides between humans. Learning from conversational instruction may be one of the most important steps toward addressing this challenge.

References

James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. Plow: A collaborative task learning agent. In *AAAI*, volume 7, pages 1514–1519.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2020. Conversational neuro-symbolic commonsense reasoning. *arXiv preprint arXiv:2006.10022*.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. 2016. Instructable intelligent personal agent. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. *arXiv preprint arXiv:1707.02363*.

J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 751–761, San Diego, California. Association for Computational Linguistics.

Richard A. Bolt. 1980. “put-that-there”: Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 14(3):262–270.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *International Conference on Learning Representations (ICLR)*.

Lera Boroditsky. 2011. How language shapes thought. *Scientific American*, 304(2):62–65.

Satchuthananthavale RK Branavan, Harr Chen, Luke S Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 82–90. Association for Computational Linguistics.

SRK Branavan, Luke S Zettlemoyer, and Regina Barzilay. 2010. Reading between the lines: Learning to map high-level instructions to commands. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1268–1277. Association for Computational Linguistics.

Crystal Chao, Maya Cakmak, and Andrea L Thomaz. 2011. Towards grounding concepts for transfer in goal learning from demonstration. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2018. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*.

Noam Chomsky. 2017. The language capacity: architecture and evolution. *Psychonomic bulletin & review*, 24(1):200–203.

John D Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, Jacob Andreas, John DeNero, Pieter Abbeel, and Sergey Levine. 2019. Guiding policies with language via meta-learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Allen Cypher and Daniel Conrad Halbert. 1993. *Watch what I do: programming by demonstration*. MIT press.

Javid Dadashkarimi, Alexander Fabbri, Sekhar Tatikonda, and Dragomir R Radev. 2018. Zero-shot transfer learning for semantic parsing. *arXiv preprint arXiv:1808.09889*.

Sanjoy Dasgupta, Daniel Hsu, Stefanos Poulis, and Xiaojin Zhu. 2019. Teaching a black-box learner. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1547–1555, Long Beach, California, USA. PMLR.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 845–854. ACM.

Biplab Deka, Zifeng Huang, and Ranjitha Kumar. 2016. Erica: Interaction mining mobile apps. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 767–776. ACM.

Barbara Di Eugenio. 1992. Understanding natural language instructions: the case of purpose clauses. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 120–127.

Mihail Eric and Christopher Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 468–473.

Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S Bernstein. 2018. Iris: A conversational agent for complex tasks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 473. ACM.

Emmanuel Ferreira, Bassam Jabaian, and Fabrice Lefèvre. 2015. Zero-shot semantic parser for spoken language understanding. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018a. Unified pragmatic models for generating and following instructions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1951–1963.

Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018b. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Dan Goldwasser and Dan Roth. 2014. Learning from natural instructions. *Machine learning*, 94(2):205–232.

Judith Good and Kate Howland. 2017. Programming language, natural language? supporting the diverse computational activities of novice programmers. *Journal of Visual Languages & Computing*, 39:78 – 92. Special Issue on Programming and Modelling Tools.

Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. 2019. Using natural language for reward shaping in reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2385–2391. International Joint Conferences on Artificial Intelligence Organization.

Mark Halpern. 1966. Foundations of the case for natural-language programming. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, AFIPS '66 (Fall), pages 639–649, New York, NY, USA. ACM.

Jonathan Herzig and Jonathan Berant. 2018. Decoupling structure and lexicon for zero-shot semantic parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1619–1629.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 851–861.

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM.

Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. 2019. Hierarchical decision making by generating and following natural language instructions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Thanapong Intharah, Daniyar Turmukhambetov, and Gabriel J Brostow. 2019. Hilc: Domain-independent pbd system via computer vision and follow-up questions. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 9(2-3):16.

Michael Janner, Karthik Narasimhan, and Regina Barzilay. 2018. Representation learning for grounded spatial reasoning. *Transactions of the Association for Computational Linguistics*, 6:49–61.

Joohyun Kim and Raymond Mooney. 2013. Adapting discriminative reranking to grounded language learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 218–227, Sofia, Bulgaria. Association for Computational Linguistics.

James R Kirk and John E Laird. 2014. Interactive task learning for simple games. *Advances in Cognitive Systems*, 3:13–30.

James R Kirk and John E Laird. 2019. Learning hierarchical symbolic representations to support interactive task learning and knowledge transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6095–6102. AAAI Press.

Amy J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. 2011. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3):21.

Igor Labutov, Shashank Srivastava, and Tom Mitchell. 2018. Lia: A natural language programmable personal assistant. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 145–150.

John E Laird, Kevin Gluck, John Anderson, Kenneth D Forbus, Odest Chadwicke Jenkins, Christian Lebiere, Dario Salvucci, Matthias Scheutz, Andrea Thomaz, Greg Trafton, et al. 2017. Interactive task learning. *IEEE Intelligent Systems*, 32(4):6–21.

John E Laird and Shiwali Mohan. 2018. Learning fast and slow: Levels of learning in general autonomous intelligent agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

John E Laird, Allen Newell, and Paul S Rosenbloom. 1987. Soar: An architecture for general intelligence. *Artificial intelligence*, 33(1):1–64.

Vu Le, Sumit Gulwani, and Zhendong Su. 2013. Smartsynth: Synthesizing smartphone automation scripts from natural language. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 193–206. ACM.

Gilly Leshed, Eben M Haber, Tara Matthews, and Tessa Lau. 2008. Coscripter: automating & sharing how-to knowledge in the enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1719–1728. ACM.

Toby Jia-Jun Li, Amos Azaria, and Brad A Myers. 2017. Sugilite: creating multimodal smartphone automation by demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6038–6049. ACM.

Toby Jia-Jun Li, Igor Labutov, Xiaohan Nancy Li, Xiaoyi Zhang, Wenze Shi, Wanling Ding, Tom M Mitchell, and Brad A Myers. 2018a. Appinite: A multi-modal interface for specifying data descriptions in programming by demonstration using natural language instructions. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 105–114. IEEE.

Toby Jia-Jun Li, Igor Labutov, Brad A Myers, Amos Azaria, Alexander I Rudnicky, and Tom M Mitchell. 2018b. Teaching agents when they fail: End user development in goal-oriented conversational agents. In *Studies in Conversational UX Design*, pages 119–137. Springer.

Toby Jia-Jun Li, Marissa Radensky, Justin Jia, Kirielle Singarajah, Tom M. Mitchell, and Brad A. Myers. 2019. Pumice: A multi-modal agent that learns concepts and conditionals from natural language and demonstrations. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, pages 577–589, New York, NY, USA. ACM.

Toby Jia-Jun Li and Oriana Riva. 2018. Kite: Building conversational bots from mobile apps. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 96–109. ACM.

Henry Lieberman. 2001. *Your wish is my command: Programming by example*. Morgan Kaufmann.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018a. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of NAACL-HLT*, pages 2060–2069.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, and Percy Liang. 2018b. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations*.

Thomas F Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomir Mech, and Ranjitha Kumar. 2018c. Learning design semantics for mobile apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pages 569–579. ACM.

Zhichu Lu, Forough Arabshahi, Igor Labutov, and Tom Mitchell. 2019. Look-up and adapt: A one-shot semantic parser. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1129–1139.

J Luketina, N Nardelli, G Farquhar, J Foerster, J Andreas, E Grefenstette, S Whiteson, and T Rocktäschel. 2019. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, August 10-16 2019, Macao, China.*, volume 57. AAAI Press (Association for the Advancement of Artificial Intelligence).

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4.

Richard G McDaniel and Brad A Myers. 1998. Building applications using only demonstration. In *Proceedings of the 3rd international conference on Intelligent user interfaces*, pages 109–116. ACM.

Aaron Mininger and John E Laird. 2018. Interactively learning a blend of goal-based and procedural tasks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.

Tom M Mitchell, John Allen, Prasad Chalasani, John Cheng, Oren Etzioni, Marc Ringuette, and Jeffrey C Schlimmer. 1991. Theo: A framework for self-improving systems. *Architectures for intelligence*, pages 323–355.

Shiwali Mohan and John Laird. 2014. Learning goal-oriented hierarchical tasks from situated interactive instruction. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Shiwali Mohan, Aaron H Mininger, James R Kirk, and John E Laird. 2012. Acquiring grounded representations of words with situated interactive instruction. In *Advances in Cognitive Systems*. Citeseer.

Brad A Myers. 1987. Creating interaction techniques by demonstration. *IEEE Computer Graphics and Applications*, 7(9):51–60.

Brad A Myers, Andrew J Ko, Chris Scaffidi, Stephen Oney, YoungSeok Yoon, Kerry Chang, Mary Beth Kery, and Toby Jia-Jun Li. 2017. Making end user development more natural. In *New Perspectives in End-User Development*, pages 1–22. Springer.

Panupong Pasupat, Tian-Shun Jiang, Evan Liu, Kelvin Guu, and Percy Liang. 2018. Mapping natural language commands to web elements. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4970–4976, Brussels, Belgium. Association for Computational Linguistics.

David Price, Ellen Rilofff, Joseph Zachary, and Brandon Harvey. 2000. Naturaljava: a natural language interface for programming in java. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 207–211. ACM.

Hilke Reckman, Jeff Orkin, and Deb Roy. 2010. Learning meanings of words and constructions, grounded in a virtual game. *g Semantic Approaches in Natural Language Processing*, page 67.

Aaron Reed. 2010. *Creating Interactive Fiction with Inform 7*, 1st edition. Course Technology Press, Boston, MA, United States.

Shashank Srivastava, Amos Azaria, and Tom M Mitchell. 2017a. Parsing natural language conversations using contextual cues. In *IJCAI*, pages 4089–4095.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017b. Joint concept learning and semantic parsing from natural language explanations. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1527–1536.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*.

Lev Semenovich Vygotsky. 1964. Thought and language. *Annals of Dyslexia*, 14(1):97–98.

Sida I Wang, Samuel Ginn, Percy Liang, and Christopher D Manning. 2017. Naturalizing a programming language via interactive learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–938.

Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2368–2378.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638.

TH Wen, D Vandyke, N Mrkšíc, M Gašíc, LM Rojas-Barahona, PH Su, S Ultes, and S Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.