Improved Models and Queries for Grounded Human-Robot Dialog

Aishwarya Padmakumar
University of Texas at Austin
aish@cs.utexas.edu

Doctoral Dissertation Proposal

Abstract

The ability to understand and communicate in natural language can make robots much more accessible for naive users. Environments such as homes and offices contain many objects that humans describe in diverse language referencing perceptual properties. Robots operating in such environments need to be able to understand such descriptions. Different types of dialog interactions with humans can help robots clarify their understanding to reduce mistakes, and also improve their language understanding models, or adapt them to the specific domain of operation.

We present completed work on jointly learning a dialog policy that enables a robot to clarify partially understood natural language commands, while simultaneously using the dialogs to improve the underlying semantic parser for future commands. We introduce the setting of opportunistic active learning - a framework for interactive tasks that use supervised models. This framework allows a robot to ask diverse, potentially off-topic queries across interactions, requiring the robot to trade-off between task completion and knowledge acquisition for future tasks. We also attempt to learn a dialog policy in this framework using reinforcement learning

We propose a novel distributional model for perceptual grounding, based on learning a joint space for vector representations from multiple modalities. We also propose a method for identifying more informative clarification questions that can scale well to a larger space of objects, and wish to learn a dialog policy that would make use of such clarifications.

Contents

Chapter 1

Introduction

The ability to understand and communicate in natural language can make robots much more accessible for naive users. This would minimally require a robot to be able to understand high level natural language commands, and detect and indicate when it has failed to understand what a user requires. Further, it would be desirable if a robot could engage in a dialog with the user, clarifying their intentions in case of uncertainty. Also, since any environment could have domain specific language (nicknames used for people, special objects such as a stethoscope in a hospital or a printer in an office), and since the robot's models may be limited in coverage, general purpose robots need to able to improve their models through interaction with users in their operating environment.

An example of a command that a service robot in an office environment may need to understand could be - "*Bring the blue mug from Alice's office*". Understanding this requires many types of capabilities. It would need to understand compositionality - how meanings of individual words combine to give meanings of phrases. Here, it would need to know that "*Alice's office*" means an office that is owned by Alice, and that "*the blue mug*" is something which is both a mug and is blue. Semantic parsing is the process of translating natural language utterances into compositional meaning representations understandable to the robot.

The robot needs to understand the meanings of words and phrases grounded in its environment. Here, it would need to know that "*Alice's office*" refers to a physical location. It would also need to be able to identify mugs and blue objects. Some knowledge, such as which office belongs to whom, can be hard coded as facts. However, environments such as homes and offices typically contain a large number of smaller objects such as mugs, whose existence and properties would be tedious to catalog. Thus a robot would need to be able to identify such objects through perception, perhaps using a camera or by manipulating it with an arm. For this, it would need to ground words such as "*blue*" and "*mug*" as perceptual properties.

If a robot only partially understands a command given to it, it would be desirable if it can ask clarification questions, such as "*What do you want me to bring?*", to obtain the missing information and avoid making mistakes. Also, if it does not know the meaning of a word such as "*mug*", it could ask the user to show examples of mugs nearby before it goes to Alice's office so that it knows what to look for. It could also opportunistically ask the user to show examples of other things, such as a "*book*", so that it is better prepared to help a different user that may need a book to be fetched. We would like the robot to be able to learn both when to ask such questions, and which questions to ask, through interactions with users.

Following a discussion of related work (chapter 2), we present completed work on jointly learning a dialog policy that enables a robot to ask clarifications when it does not fully understand natural language commands, while simultaneously using the dialogs to improve the underlying semantic parser for future commands (chapter 3). We also present the framework of opportunistic active learning in the context of understanding natural language descriptions of objects. We demonstrate its effectiveness in this task (chapter 4), and present work on learning a dialog policy for choosing such queries (chapter 5).

In proposed work (chapter 6), we discuss a new model for perceptual grounding, based on learning a joint space for vector representations from multiple modalities (6.1). We also propose a method for identifying more informative clarification questions that can scale well to a larger space of objects (section 6.2), and wish to learn a dialog policy that would make use of such clarifications (section 6.3).

Chapter 2

Background and Related Work

2.1. Natural language commands to Robots

Communicating with robots in natural language has been an area of interest for a long time. Theobalt et al. (2002) is an early attempt at developing a dialog system interface over low level navigation. Users could query the robot about it's position and command it to navigate to a specific location using rich language including landmarks. Semantic parsing with hand-coded rules was used for language understanding. Another early work is that of Lauria et al. (2002) which proposes a learning from demonstration framework that uses natural language to instruct robots at a symbolic level. These used pre-programmed language understanding, perceptual grounding and dialog policies. They also do not evaluate the performance of the system.

She et al. (2014) build a dialog system for instructing robots to combine simple instructions in a blocks world to complex ones. They also use fixed components for semantic parsing, perceptual grounding and the dialog policy, but report success rates for teaching different types of instructions. In a contemporary work, Matuszek et al. (2013) learn a semantic parser from paired sentences and annotated semantic forms to map natural language commands to high level goals that are more independent of the environment. A related work is that of Chen and Mooney (2011) who learn a semantic parser to translate natural language route instructions to logical plans for a simulated environment.

Other works use fixed components for parsing natural language but learn models for grounding to symbols in the robot's knowledge base. Kollar et al. (2013b) use a probabilistic model to perform grounding, and use a static dialog policy to add new concepts to the knowledge base used for grounding. Tellex et al. (2014) develop a graphical model for grounding that makes use of a pretrained parser. The model can also be used for generating clarification questions. Arumugam et al. (2018) learn a probabilistic model for grounding to symbolic goals that act as rewards in a hierarchical state and action MDP space to handle commands of varying levels of abstraction. Also related is Bastianelli et al. (2016), who use symbolic grounding to resolve ambiguities from semantic parsing.

Other works focus on learning dialog policies for effective human-robot communication. Zhang and Stone (2015) develop a system that learns a dialog policy by modeling dialog as a POMDP (section 2.3.2). They also incorporate logical reasoning, and common sense knowledge with the result of pretrained natural language understanding components. Whitney et al. (2017) learn a policy for clarification dialogs that can incorporate both natural language responses and gestures made by users.

Human-robot dialog can also be used to improve the natural language understanding system used by the robot. Thomason et al. (2015) develop a system that learns a semantic parser for understanding natural language commands. They use a static policy to ask clarification questions, but also use responses from clarifications as weak supervision to improve the parser. We extend this in our work (chapter 3) by learning a dialog policy for clarifications while simultaneously improving the parser from clarification responses.

More recently, there has been interest in learning end-to-end neural networks to map natural language instructions and observations directly to action sequences. Earlier work in this space was either on tasks that require only grounding to a knowledge base, such as mapping to formal queries (Suhr et al., 2018), or used simulated datasets that did not require real perception (Mei et al., 2016; Misra et al., 2017a). Recently, large scale simulated datasets (Chang et al., 2017; Yan et al., 2018) have enabled the development of end to end neural networks that use complex visual observations to map natural language commands to actions for tasks such as following route instructions (Anderson et al., 2018), embodied question answering (Das et al., 2018), navigation combined with object manipulation (Misra et al., 2017a), and continuous control of a quadcopter drone (Blukis et al., 2018).

2.2. Semantic Parsing

Semantic parsing maps a natural language sentence such as "Go to Alice's office" to a machine understandable meaning representation. In our work, we use λ-calculus logical forms such as:

$$\texttt{navigate}(\texttt{the}(\lambda x.\texttt{office}(x) \wedge \texttt{possess}(\texttt{alice}, x) \wedge \texttt{person}(\texttt{alice})))$$

This represents that the robot should navigate to a place x which is an office, and owned by a person `alice`.

This formalism reduces the number of lexical entries the system needs to learn by exploiting compositional reasoning over language. For example, if it learns that "Alice Ashcraft" also refers to the entity `alice`, it does not need to learn another lexical entry for "Alice Ashcraft's office".

There has been considerable work in semantic parsing using direct supervision in the form of annotated meaning representations (Wong and Mooney, 2007; Kwiatkowski et al., 2013; Berant et al., 2013). More recent works use indirect signals from downstream tasks. Artzi and Zettlemoyer (2011) use clarification dialogs to train semantic parsers for an airline reservation system without explicit annotation of meaning representations. Thomason et al. (2015), incorporate this general approach into a system for instructing a mobile robot using a basic dialog state and fixed hand-coded policy.

In our work (chapter 3), semantic parsing is performed using probabilistic CKY-parsing with a Combinatory Categorial Grammar (CCG) (Steedman and Baldridge, 2011) and meanings associated with lexical entries (Zettlemoyer and Collins, 2005). Perceptron-style updates to parameter values are used during training to weight parses to speed search and give confidence scores in parse hypotheses.

2.3. Reinforcement Learning

Reinforcement learning is a computational process of learning to map situations to actions to maximize a numerical reward signal (Sutton et al., 1998). In a reinforcement learning problem, an agent interacts with its environment to achieve a goal. The agent can sense the state of the environment, take actions that affect the state, and have one or more goals related to this state. It typically needs to take a sequence of actions to achieve its goal and may receive only delayed numerical feedback (reward) to indicate whether progress has been made.

Reinforcement learning faces the challenge of trading off exploration and exploitation. The agent has to *exploit* actions known to be effective, to obtain reward, but must *explore* new actions to find out those that are the most effective.

We now discuss two common formulations of reinforcement learning problems (sections 2.3.1 and 2.3.2), and three algorithms used for policy learning (sections 2.3.3, 2.3.4 and 2.3.5).

2.3.1. Markov Decision Process (MDP)

A Markov Decision Process (MDP) is a tuple $\langle \mathbb{S}, \mathbb{A}, \mathbb{T}, \mathbb{R}, \gamma \rangle$, $\mathbb{S}$ is a set of states, $\mathbb{A}$ is a set of actions, $\mathbb{T}$ is a transition function, $\mathbb{R}$ is a reward function and γ is a discount factor. At any instant of time t, the agent is in a state $s_t \in \mathbb{S}$. It chooses to take an action $a_t \in \mathbb{A}$ according to a policy π, commonly represented as a probability distribution over actions where $\pi(a_t|s_t)$ is the probability of taking action a_t when the agent is in state s_t. On taking action a_t, the agent is given a real-valued reward r_t and transitions to a state s_{t+1}.

State transitions occur according to the probability distribution $P(s_{t+1}|s_t, a_t) = \mathbb{T}(s_t, a_t, s_{t+1})$, and rewards obtained follow the distribution $P(r_t|s_t, a_t) = \mathbb{R}(s_t, a_t, s_{t+1})$.

The objective is to identify a policy π that is optimal in the sense that it maximizes the expected long term discounted reward, called return, given by

$$g = \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^t r_t \right]$$

2.3.2. Partially Observable Markov Decision Process (POMDP)

A Partially Observable Markov Decision Process (POMDP) is an extension of MDPs where the agent does not know what state it is in, but only receives a noisy observation indicative of the state.

Formally, a POMDP is a tuple $(\mathbb{S}, \mathbb{A}, \mathbb{T}, \mathbb{R}, \mathbb{O}, \mathbb{Z}, \gamma, b_0)$, where $\mathbb{S}$ is a set of states, $\mathbb{A}$ is a set of actions, $\mathbb{T}$ is a transition function, $\mathbb{R}$ is a reward function, $\mathbb{O}$ is a set of observations, $\mathbb{Z}$ is an observation function, γ is a discount factor and b_0 is an initial belief state (Kaelbling et al., 1998).

These are defined as in MDPs, but the the state s_t is hidden from the agent and only a noisy observation $o_t \in \mathbb{O}$ of s_t is available to it. The agent maintains a belief state b_t which is a distribution over all possible states it could be in at time t. $b_t(s_i)$ gives the probability of being in state s_i at time t. The agent chooses actions $a_t \in \mathbb{A}$ based on b_t, according to a policy π. On taking action a_t, the agent is given a real-valued reward r_t, transitions to a state s_{t+1}, and receives a noisy observation o_{t+1} of s_{t+1}, which is used to update its belief b_{t+1}.

State transitions occur according to the probability distribution $P(s_{t+1}|s_t, a_t) = \mathbb{T}(s_t, a_t, s_{t+1})$, observations are related to the states by the probability distribution $P(o_t|s_t, a_{t-1}) = \mathbb{Z}(o_t, s_t, a_{t-1})$ and rewards obtained follow the distribution $P(r_t|s_t, a_t) = \mathbb{R}(s_t, a_t, s_{t+1})$.

The objective, again, is to identify a policy π that is optimal in the sense that it maximizes return.

2.3.3. REINFORCE Algorithm

The REINFORCE algorithm (Williams, 1992) is a simple policy gradient algorithm used to learn a policy in an MDP. The agent learns a policy $\pi(a|s;\theta)$, parameterized with weights θ that computes the probability of taking action a in state s. An example is a policy based on a feature representation $f(s,a)$ for a state-action pair (s,a):

$$\pi(a|s;\theta) = \frac{e^{\theta^T f(s,a)}}{\sum_{a'} e^{\theta^T f(s,a')}}$$

where the denominator is a sum over all actions possible in state s.

The weights are updated using a stochastic gradient ascent rule:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

where $J(\theta)$ is the expected return from the policy according to the distribution over trajectories induced by the policy.

2.3.4. Q-Learning

The quality of a policy π can be estimated using the action value function

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a \right]$$

The optimal policy satisfies the Bellman equation,

$$Q^*(s,a) = \mathbb{E}_{s'} \left[\mathbb{R}(s,a,s') + \gamma max_{a' \in \mathbb{A}} Q^*(s',a') \right]$$

For a POMDP, the above equations would be in terms of belief states b.

Q-learning is a temporal difference method used for policy learning. The algorithm starts off with possibly arbitrary estimates for $Q(s,a)$ and attempts to update them towards $Q^*(s,a)$ through experience. This experience can be collected using any policy, and hence, the algorithm is an off-policy algorithm. The following update is performed when the agent takes action a_t in state s_t, receiving reward r_t, and transitioning to s_{t+1}.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q(s_{t+1}, a'))$$

Given the final estimates $\hat{Q}(s,a)$, the corresponding learned policy would be to take the action fo highest estimated value at each state. That is,

$$\pi(a|s) = \max_a Q(s,a)$$

2.3.5. KTD-Q Learning

When the state space is very large or continuous, Q^π cannot be computed for each state (or belief state) individually and is hence assumed to be a function with parameters θ over some features that represent the state. When the transition or reward dynamics are not constant (*non-stationary* problem), a suitable approximation is the Kalman Temporal Differences framework (Geist and Pietquin, 2010). This casts the function approximation as a filtering problem and solves it using Kalman filtering. The specialization for learning the optimal action value function is called the KTD-Q algorithm.

Filtering problems estimate hidden quantities X from related observations Y, modeling X and Y as random variables. When estimating action values, X corresponds to the function parameters, θ and the observations are the estimated returns, $r_t + \gamma \max_a \hat{Q}_{\theta_t}(s_{t+1}, a)$, and a random noise is added to both of these to allow for parameters to change over time. The update rules are derived from Kalman Filtering Theory and details can be found in Geist and Pietquin (2010).

2.4. Dialog Systems

Spoken Dialog Systems allow users to interact with information systems with speech as the primary form of communication (Young et al., 2013). They were originally deployed for call center operations such as airline ticket reservation (Hemphill et al., 1990), and restaurant recommendation (Wen et al., 2017). More recently, dialog systems have become popular for issuing simple commands on mobile phones through virtual assistants such as Apple's Siri, Google Voice and Amazon's Alexa.

Spoken dialog systems typically follow a pipeline similar to that in figure 2.1. The user utterance is first processed by a speech recognition module, which produces a text transcript. This is followed by a language understanding module that extracts the information provided by the user in the utterance. This is used by the dialog state tracking module to update the system's belief of what the user wishes to accomplish from the interaction. Following this, the dialog management module uses the system's dialog policy to decide which dialog action to take next, for example ask for more information. The response generation module converts this abstract dialog act into a natural language response, which is rendered into speech by the speech synthesis module.

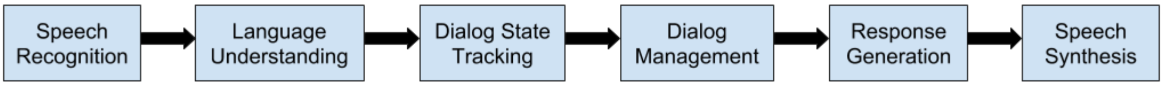


Figure 2.1: Spoken Dialog System Pipeline

There has been considerable research in goal directed dialog systems targeted at performing call-center type tasks (Young et al., 2013). These systems model dialog as a POMDP and focus on either the problem of tracking belief state accurately over the large state spaces (Young et al., 2010; Thomson and Young, 2010; Mrkšić et al., 2015) or that of efficiently learning a dialog policy over such state spaces (Gašić and Young, 2014; Pietquin et al., 2011). These systems typically assume that the other components of the pipeline are fixed. Some of our completed work (chapter 3) combines this research with research on learning semantic parsers from weak supervision provided by clarification dialogs.

More recently, there has been work on modeling various components of a dialog system using neural networks (Mrkšić et al., 2015; Wen et al., 2015a). There have also been some end-to-end neural network systems that simultaneously learn dialog policy and language comprehension (Wen et al., 2017; Williams and Zweig, 2016; Bordes and Weston, 2016). A major challenge in these systems is to find database entries satisfying certain constraints, and ensuring that all relevant information is included in the system's responses. Some systems assume that these functions are performed by deterministic APIs (Bordes and Weston, 2016). Others attempt to design neural networks to perform these functions (Wen et al., 2015b; Kiddon et al., 2016).

Some other tasks for which dialog systems have been developed are open domain conversations (Serban et al., 2016), playing 20 questions games about famous people (Hu et al., 2018), and converting natural language to code snippets (Chaurasia and Mooney, 2017).

2.5. Active Learning

In machine learning tasks where obtaining labeled examples is expensive, active learning is used to lower the cost of annotation without sacrificing model performance. Active learning allows a learner to iteratively query for labels of unlabeled data points that are expected to maximally improve the existing model. Research in active learning attempts to identify examples that are likely to be the most useful in improving a supervised model. A number of metrics have been proposed to evaluate examples, including uncertainty sampling (Lewis and Gale, 1994), density-weighted methods (Settles and Craven, 2008), expected error reduction (Roy and McCallum, 2001), query by committee (Seung et al., 1992), and the presence of conflicting evidence (Sharma and Bilgic, 2016); as surveyed by Settles (2010).

Multilabel active learning is the application of active learning in scenarios where multiple labels, that are not necessarily mutually exclusive, are associated with a data point (Brinker, 2006). These setups often suffer from sparsity, both in the number of labels that are positive for a data point, and in the number of positive data points per label. Standard active learning metrics are often extended to the multilabel setting, by assuming that one-vs-all classifiers are learned for each label, and that all the learned classifiers are comparable (Brinker, 2006; Singh et al., 2009; Li et al.). Label statistics have also been incorporated into heuristics for selecting instances to be queried (Yang et al., 2009; Li and Guo, 2013). There have also been Bayesian approaches that select both an instance and label to be queried (Qi et al., 2009; Vasisht et al., 2014).

The most commonly used framework for active learning is pool-based active learning, where the learner has access to the entire pool of unlabeled data at once, and can iteratively query for examples. In contrast, sequential active learning is a framework in which unlabeled examples are presented to the learner in a stream (Lewis and Gale, 1994). For every example, the learner can decide whether to query for its label or not. This results in an additional challenge – since the learner cannot compare all unlabeled data points before choosing queries, each query must be chosen based on local information only. We introduce the framework of Opportunistic Active Learning (chapter 4) that extends sequential active learning to an interactive multi-label task.

Recently, there has been interest in using reinforcement learning to learn a policy for active learning. Fang et al. (2017) use deep Q-learning to acquire a policy that sequentially examines unlabeled examples and decides whether or not to query for their labels; using it to improve named entity recognition in low resource languages. Also, Bachman et al. (2017) use meta-learning to jointly learn a data selection heuristic, data representation and prediction function for a distribution of related tasks. They apply this to one shot recognition of characters from different languages, and in recommender systems. Woodward and Finn (2017) use reinforcement learning with a recurrent-neural-network-based Q-function in a sequential one-shot learning task to decide between predicting a label and acquiring the true label at a cost. We follow in this line of work to learn a policy for opportunistic active learning in a task of grounding natural language descriptions of objects (chapter 5).

2.6. Human-robot Dialog for Teaching Perceptual Concepts

A first step towards teaching robots perceptual concepts through dialog is Kollar et al. (2013a), who develop a system that uses semantic parsing for language understanding, and grounds meanings of words using SVM-based perceptual classifiers. This is trained using pairs of images and corresponding language descriptions, and can generate descriptions of objects, but these not evaluated as a complete interactive system.

Other works combine these capabilities to perform clarifications to better ground descriptions at test time (Dindo and Zambuto, 2010; Parde et al., 2015). Vogel et al. (2010) learn to ground simple perceptual concepts using only a 20-questions style game, where there is no initial description, but learning is entirely driven by the robot's queries of whether a concept applies to an object. Kulick et al. (2013) use active learning to enable a robot to learn spatial relations by manipulating objects into specific positions and querying an oracle about whether a relation holds. However they receive ground truth positions of objects and do not perform perception.

Thomason et al. (2016) demonstrate that multimodal perceptual concepts, with richer visual features, as well as auditory and haptic features, can be learned from an I Spy game, by pairing descriptions with correct guesses by the robot.

Some works also try to learn a dialog policy for learning new ways to refer to known perceptual concepts (Yu et al., 2017). The perceptual concepts are basic, and dialogue policy is learned through reinforcement learning from a dataset of human-human conversations. Yu et al. (2016) demonstrate the importance of taking initiative, processing and expressing perceptual concepts, and understanding ellipsis for the same task.

More generally, natural language can be used to aid learning from demonstration. She and Chai (2017) learn a system that uses a hierarchical knowledge base over actions to compose simple action primitives into complex ones. This is learned from human demonstrations paired augmented by language descriptions, where the robot learns a dialog policy to ask clarifications for noun phrase grounding, effects of an action, states of objects and whether actions are necessary to achieve a goal.

In our completed work (chapter 4), we introduce the setting of opportunistic active learning - a framework for interactive tasks that involve learning of supervised models. This framework allows a robot to ask more diverse queries across interactions, and requires the robot to trade-off between task completion and knowledge acquisition for future tasks.

2.7. Grounding Language in Perception

When humans interact with robots in natural language, they typically refer to entities in the real world, and expect robots to be able to identify these referents. For many types of entities, such as physical household objects, people typically describe them in terms of attributes such as object category, color and weight (Guadarrama et al., 2016; Thomason et al., 2016). Robots need to be able to perceive properties that humans refer to, and use these to map referring expressions to referents in the real world. This task is an instance of the symbol grounding problem (Harnad, 1990).

There has been considerable work on extending word representations to incorporate visual context. These are found to be useful for predicting lexical similarity (Silberer and Lapata, 2012, 2014; Lazaridou et al., 2015a), verifying common sense assertions (Kottur et al., 2016), verifying visual paraphrasing (Kottur et al., 2016), image categorization (Silberer and Lapata, 2014; Lazaridou et al., 2015a) including in the zero shot setting (Lazaridou et al., 2014), and retrieval of related images (Kottur et al., 2016). However, these do not attempt to retrieve images or objects based on free-form natural language descriptions - as desired in robotics applications.

Guadarrama et al. (2016) assemble a dataset for retrieval of objects based on open vocabulary natural language descriptions, and compare the performance of image categorization and instance recognition methods, as well as ensembles of these on this task. Misra et al. (2017b) learn a network to more intelligently compose classifiers learned for adjectives and nouns. Hu et al. (2016) propose a neural network model that uses a vector representation of a region crop, the entire image, and relative bounding box coordinates to score regions in an image to identify the one referred to by a natural language expression. Other works either align vector representations of images and descriptions/ captions using methods such as CCA (Feng et al., 2015), or learn a joint embedding of the modalities (Wang et al., 2016) to perform image-to-caption and caption-to-image retrieval. Xiao et al. (2017) learn to ground descriptions in images by learning mappings from phrases to attention vectors over the image, and combining attended regions using linguistic constraints. These works each set up the grounding problem in different ways, and evaluate their methods on different datasets. We propose to perform a comparison of these different types of methods for language grounding.

We also propose a new method based on learning a joint embedding of language and visual modalities (section 6.1) that uses a loss function based on retrieval (Wang et al., 2016) but using word representations instead of phrase/ sentence representations (Silberer and Lapata, 2014; Lazaridou et al., 2015a; Kottur et al., 2016) for better generalization. We also propose to compare this with other proposed methods for grounding based on classifiers (Guadarrama et al., 2016), a direct scoring network (Hu et al., 2016) and previous distributional approaches (Wang et al., 2016).

There are also works that focus on learning other aspects of grounding including spatial relations (Bisk et al., 2016), relative properties such as size, weight and rigidity of object pairs (Forbes and Choi, 2017), subject-relation-object triples (Hu et al., 2017), meanings of verbs modeled as state changes (Gao et al., 2016; Liu et al., 2016; Gao et al., 2018), and semantic roles of a verb in videos (Yang et al., 2016). Grounding of object descriptions can also be improved by incorporating information such as temporal context and gesture (Williams et al., 2017).

While most work on grounded language learning focuses on understanding, there is also work on generating referring expressions of objects for effective human-robot communication (Fang et al., 2013, 2014).

2.8. Visually Grounded Dialog

Recently, a few new dialog tasks and datasets have been introduced that require grounding of language in images. The VidDial dataset was collected to teach a robot to coherently answer a sequence of questions about a single image (Das et al., 2017a). This has also been used to train a pair of agents, one of which asks questions about an unseen image, and another that answers them using the image (Das et al., 2017b), with the goal of the questioning agent attempting to learn a representation of what the image looks like. They find that the answering agent does provide answers similar to humans, and only pretraining constrains the agents to retain the semantics of English as used by humans.

Another related work is the GuessWhat?! dataset (De Vries et al., 2017) of humans playing a 20 questions game to identify an object in images of rich scenes. This is used to train a questioner that tries to identify the target object by asking yes/no questions, and the oracle that learns to answer them. It is difficult to evaluate the success of either agent, as ground truth for both agents is unlikely to be present in the training set. Further, if they are trained jointly, the challenge of retaining human semantics again arises. Another work that learns to ask questions that can discriminate between images is Li et al. (2017).

We propose to learn a system that can learn clarification questions that can refine on an initial description (sections 6.2 and 6.3) - which is not present in the above tasks. Further, we wish to do this in a setting where we can provide ground truth answers to questions of the system during training. Hence we choose a more restrictive set of questions, and propose to answer them using annotations of objects and attributes as in our completed work (chapter 5).

Chapter 3

Integrated Learning of Dialog Strategies and Semantic Parsing

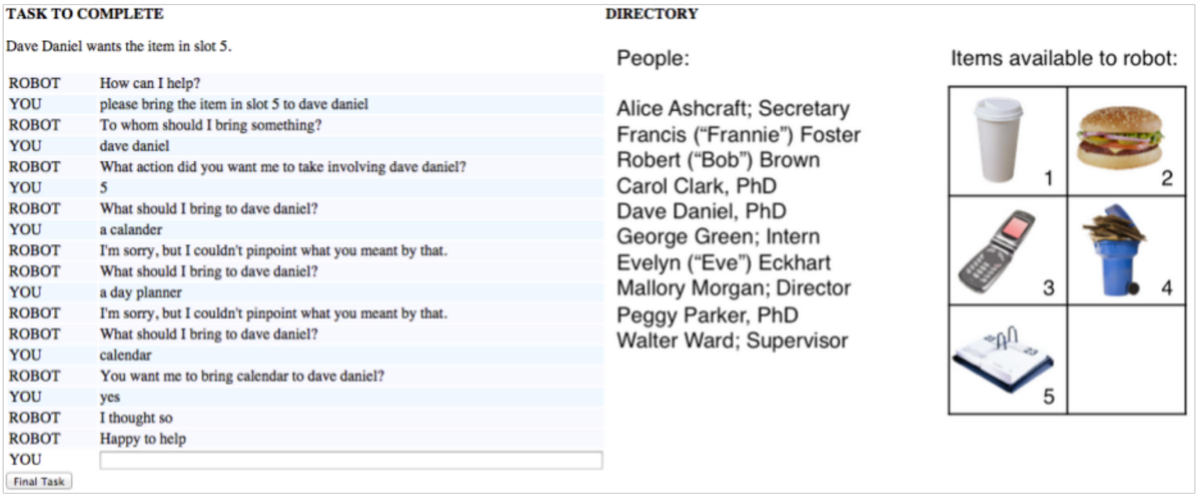
Robots need to be able to understand high-level natural language commands to be accessible to naive users. Since the types of commands, and language usage vary across domains, it is desirable that a robot should be able to improve through interaction with users in its operating environment. For an interactive dialog system, prior work had demonstrated different methods to independently improve either the natural language understanding component or the dialog strategy. We discuss an approach to integrate the learning of both a dialog strategy using reinforcement learning, and a semantic parser for robust natural language understanding, using only natural dialog interaction for supervision. Full details are available in Padmakumar et al. (2017). The main challenge involved is choosing an appropriate reinforcement learning algorithm, and training procedure, as the simultaneous training of the parser violates the assumption of a non-stationary environment, made by most reinforcement learning algorithms.

3.1. Task and System Setup

Our goal is to develop a dialog system that enables a robot to translate high level commands such as *"go to Alice's office"* to a command represented as an action with associated semantic roles. The user can command the system to perform two actions: navigation and delivery. The robot makes an initial guess of the desired action from the user's response, and then may ask clarification questions in case of insufficient understanding. At each step, it can respond with one of four dialog acts: asking the user to repeat their command, confirming a command or an argument value, requesting a specific argument of a command, and executing an action (thereby ending the dialog). A sample dialog is shown in figure 3.1.

Given a user response, a semantic parser is used to convert utterances to logical forms (section 2.2) which are then grounded to actions and semantic roles using world knowledge. The robot maintains a belief state that represents the probability that a particular combination of action and semantic roles is the command intended by the user. This is updated based on the grounded response, and is then used by the dialog policy to choose the next action to be taken by the robot. Once the conversation is complete, the parser and policy can be updated from the dialog (sections 3.2 and 3.3 respectively).

The dialog is considered a success if the final action taken is correct and a failure oth-



TASK TO COMPLETE

Dave Daniel wants the item in slot 5.

ROBOT	How can I help?
YOU	please bring the item in slot 5 to dave daniel
ROBOT	To whom should I bring something?
YOU	dave daniel
ROBOT	What action did you want me to take involving dave daniel?
YOU	5
ROBOT	What should I bring to dave daniel?
YOU	a calander
ROBOT	I'm sorry, but I couldn't pinpoint what you meant by that.
ROBOT	What should I bring to dave daniel?
YOU	a day planner
ROBOT	I'm sorry, but I couldn't pinpoint what you meant by that.
ROBOT	What should I bring to dave daniel?
YOU	calendar
ROBOT	You want me to bring calendar to dave daniel?
YOU	yes
ROBOT	I thought so
ROBOT	Happy to help
YOU	

Final Task

DIRECTORY

People:

Alice Ashcraft; Secretary
Francis ("Frannie") Foster
Robert ("Bob") Brown
Carol Clark, PhD
Dave Daniel, PhD
George Green; Intern
Evelyn ("Eve") Eckhart
Mallory Morgan; Director
Peggy Parker, PhD
Walter Ward; Supervisor

Items available to robot:

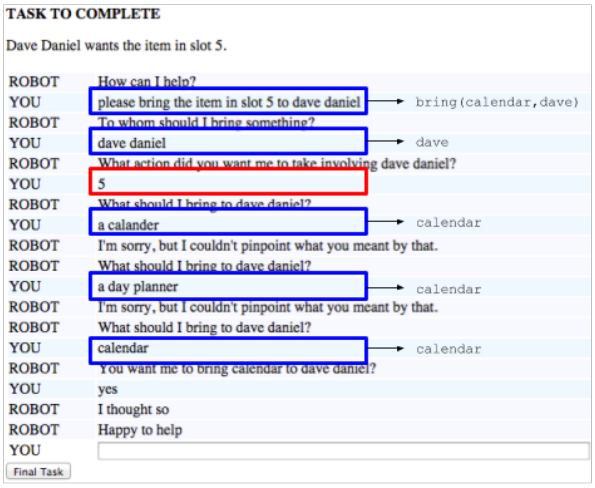
Figure 3.1: A sample dialog in our interface on Amazon Mechanical Turk.

erwise. The user also has the ability to prematurely end the dialog, and any conversation terminated in this manner is also considered a failure.

3.2. Semantic Parser Learning

The semantic parser is trained using paired sentences and logical forms. A small supervised training set is used to initialize the parser. Training continues using pairs obtained through weak supervision collected from user dialogs (Thomason et al., 2015).

Figure 3.2 shows an example of the training pairs induced from the example dialog. To obtain these, we obtain multiple semantic parses for these responses, and parses are syntactically valid, and that ground to the action finally taken by the robot or its arguments, are paired with the response to training pairs. These paired responses and semantic forms can then be used to retrain the parser between conversations. While this weak supervision may be noisy, the syntactic and grounding constraints remove most spurious examples.



TASK TO COMPLETE

Dave Daniel wants the item in slot 5.

ROBOT	How can I help?	
YOU	please bring the item in slot 5 to dave daniel	bring(calendar,dave)
ROBOT	To whom should I bring something?	
YOU	dave daniel	dave
ROBOT	What action did you want me to take involving dave daniel?	
YOU	5	
ROBOT	What should I bring to dave daniel?	
YOU	a calander	calendar
ROBOT	I'm sorry, but I couldn't pinpoint what you meant by that.	
ROBOT	What should I bring to dave daniel?	
YOU	a day planner	calendar
ROBOT	I'm sorry, but I couldn't pinpoint what you meant by that.	
ROBOT	What should I bring to dave daniel?	
YOU	calendar	calendar
ROBOT	You want me to bring calendar to dave daniel?	
YOU	yes	
ROBOT	I thought so	
ROBOT	Happy to help	
YOU		

Final Task

Figure 3.2: Training pairs induced from the clarification dialog, by selecting parses that ground to the final action and its arguments. The response in red is discarded because no such parse is found.

3.3. Dialog Policy Learning

We use a POMDP to model dialog and learn a policy (section 2.4), adapting the Hidden Information State model (HIS) (Young et al., 2010) to track the belief state as the dialog progresses. The key idea behind this approach is to group states into equivalence classes called partitions, and maintain a probability for each partition instead of each state. States within a partition are those that are indistinguishable to the system given the current dialog.

More concretely, our belief state can be factored into two main components - the goal intended by the user $\mathbf{g}$ and their most recent utterance $\mathbf{u}$. A partition p is a set of possible goals which are equally probable given the conversation so far.

After every user response, a beam of possible choices for $\mathbf{u}$ can be obtained by grounding the beam of top-ranked parses from the semantic parser. Grounding is performed by looking up a knowledge base of entities such as people, and relations such as who owns an office. Given the previous system action $\mathbf{m}$, The belief $b(p, \mathbf{u})$ is calculated as in the HIS model as follows

$$b(p, \mathbf{u}) = k * P(\mathbf{u}) * T(\mathbf{m}, \mathbf{u}) * M(\mathbf{u}, \mathbf{m}, p) * b(p)$$

Here, $P(\mathbf{u})$ is the probability of the utterance hypothesis $\mathbf{u}$ given the user response, which is obtained from the semantic parser. $T(\mathbf{m}, \mathbf{u})$ is a 0-1 value indicating whether the response is relevant given the previous system question, determined from the semantic type of the response. $M(\mathbf{u}, \mathbf{m}, p)$ is a 0-1 value indicating whether goals in partition p are relevant to the response and previous system question. $b(p)$ is the belief of partition p before the update, obtained by marginalizing out $\mathbf{u}$ from $b(p, \mathbf{u})$. k is a normalization constant that allows the expression to become a valid probability distribution.

We extract features from the belief state to form a summary space over which a dialog policy is learned as in prior work (Young et al., 2010; Gašić and Young, 2014). Table 3.1 contains the features used to learn the policy.

Probability of top hypothesis
Probability of second hypothesis
Number of goals allowed by the partition in the top hypothesis
Number of parameters of the partition in the top hypothesis, required by its action, that are uncertain (set to the maximum value if there is more than one possible action)
Number of dialog turns used so far
Do the top and second hypothesis use the same partition (0-1)
Type of last user utterance
Action of the partition in the top hypothesis, or *null* if this is not unique

Table 3.1: Features used in summary space

The choice of policy learning algorithm is important because learning POMDP policies is challenging and dialog applications exhibit properties not often encountered in other reinforcement learning applications (Daubigney et al., 2012). We use KTD-Q (Kalman Temporal Difference Q-learning (Geist and Pietquin, 2010)) to learn the dialog policy as it was designed to satisfy some of these properties and tested in a dialog system with simulated users (Pietquin et al., 2011). The properties we wished to be satisfied by the algorithm were the following:

- Low sample complexity in order to learn from limited user interaction.
- An off-policy algorithm to enable the use of existing dialog corpora to bootstrap the system, and crowdsourcing platforms such as Amazon Mechanical Turk during training and evaluation.
- A model-free rather than a model-based algorithm because it is difficult to design a good transition and observation model for this problem (Daubigney et al., 2012).

- Robustness to non-stationarity because the underlying language understanding component changes with time (Section 3.2), which is likely to change state transitions.

To learn the policy, we provided a high positive reward for correct completion of the task and a high negative reward when the robot chose to execute an incorrect action, or if the user terminated the dialog before the robot was confident about taking an action. The system was also given a per-turn reward of -1 to encourage shorter dialogs.

3.4. Experiments

The semantic parser was initialized using a small seed lexicon and trained on a small set of supervised examples constructed using templates. The dialog policy was initialized with an approximation of a good static policy.

3.4.1. Platform and setup

Our experiments were done through Mechanical Turk as in previous work (Thomason et al., 2015; Wen et al., 2017). The setup is shown in figure 3.1. During the training phase, each user interacted with one of four dialog agents (described in section 3.4.2), selected uniformly at random. Users were not told of the presence of multiple agents and were not aware of which agent they were interacting with. They were given a prompt for either a navigation or delivery task and were asked to have a conversation with the agent to accomplish the given task. No restrictions were placed on the language they could employ. We use visual prompts for the tasks to avoid linguistic priming (e.g. a picture of a hamburger instead of the word "hamburger"). Training dialogs were acquired in 4 batches of 50 dialogs each across all agents. After each batch, agents were updated as described in section 3.4.2.

A final set of 100 test conversations were then conducted between Mechanical Turk users and the trained agents. These test tasks were novel in comparison to the training data in that although they used the same set of possible actions and argument values, the same combination of action and argument values had not been seen at training time. For example, if one of the test tasks involved delivery of a `hamburger` to `alice`, then there may have been tasks in the training set to deliver a `hamburger` to other people and there may have been tasks to deliver other items to `alice`, but there was no task that involved delivery of a `hamburger` to `alice` specifically.

3.4.2. Dialog agents

We compared four dialog agents. The first agent performed only parser learning (described in Section 3.2). Its dialog policy was always kept the static policy used to initialize the KTD-Q algorithm. Its parser was incrementally updated after each training batch. This agent is similar to the system used by (Thomason et al., 2015) except that it uses the same state space as our other agents, and multiple hypotheses from the parser, for fairer comparison.

The second agent performed only dialog policy learning. Its parser was always kept to be the initial parser that all agents started out with. Its policy was incrementally updated after each training batch using the KTD-Q algorithm. The third agent performed both parser and dialog learning; but instead of incrementally updating the parser and policy after each batch, they were trained at the end of the training phase using dialogs across all batches. This would not allow the dialog manager to see updated versions of the parser in batches after the first and adapt the policy towards the improving parser. We refer to this as *full* learning of parser and dialog policy. The fourth agent also performed both parser and dialog learning. Its parser and policy were updated incrementally after each training batch. Thus for the next training batch, the changes due to the improvement in the parser from the previous batch could, in theory, be demonstrated in the dialogs and hence contribute towards updating the policy in a manner consistent with it. We refer to this as *batchwise* learning of parser and dialog policy.

3.4.3. Experiment hypothesis

We hypothesized that the agent performing *batchwise* parser and policy learning would outperform the agents performing only parser or only dialog learning as we expect that improving both components is more beneficial. However, we did not necessarily expect the same result from *full* parser and dialog learning because it did not provide any chance to allow updates to propagate even indirectly from one component to another, exposing the RL algorithm to a more *non-stationary* environment. Hence, we also expected *batchwise* learning to outperform *full* learning.

3.4.4. Results and Discussion

The agents were evaluated on the test set using the following objective performance metrics: the fraction of successful dialogs and the length of successful dialogs. We also included a survey at the end of the task asking users to rate on a 1–5 scale whether the robot understood them, and whether they felt the robot asked sensible questions.

Table 3.2 gives the agents' performance on these metrics. All differences in dialog success and the subjective metrics are statistically significant according to an unpaired t-test with $p < 0.05$. In dialog length, the improvement of the *batchwise* learning agent over the agents performing only parser or only dialog learning are statistically significant.

As expected, the agent performing *batchwise* parser and dialog learning outperforms the agents performing only parser or only dialog learning, in the latter case by a large margin. We believe the agent performing only parser learning performs much better than the agent performing only dialog learning due to the relatively high sample complexity of reinforcement learning algorithms in general, especially in the partially observable setting. In contrast, the parser changes considerably even from a small number of examples. Also, we observe that *full* learning of both components does not in fact outperform only parser learning. We believe this is because the distribution of hypotheses obtained using the initial parser at training time

Learning involved	% successful dialogs	Avg dialog length	Robot understood	Sensible questions
Parser	75	12.43	2.93	2.79
Dialog	59	11.73	2.55	2.91
Parser & Dialog - *full*	72	12.76	2.79	**3.28**
Parser & Dialog - *batchwise*	**78**	**10.61**	**3.30**	3.17

Table 3.2: Performance metrics for dialog agents tested. Differences in dialog success and subjective metrics are statistically significant according to an unpaired t-test with $p < 0.05$.

is substantially different from that obtained using the updated parser at test time. We believe that *batchwise* training mitigates this problem because the distribution of hypotheses changes after each batch of training and the policy when updated at these points can adapt to some of these changes. The optimal size of the batch is a question for further experimentation. Using a larger batch is less likely to overfit updates to a single example but breaking the total budget of training dialogs into more batches allows the RL algorithm to see less drastic changes in the distribution of hypotheses from the parser.

We also observe quantitative improvements in parser accuracy for agents whose parsers were trained. With dialog policy learning, a qualitative change observed is that the system tends to confirm or act upon lower probability hypotheses than is recommended by the initial hand-coded policy. This is possibly because as the parser improves, its top hypotheses are more likely to be correct.

Chapter 4

Opportunistic Active Learning for Grounding Natural Language Descriptions

An important skill required by robots in a home or office setting is retrieving objects based on natural language descriptions. We find a number of objects such as books, mugs and bottles in such environments, that users typically refer to using a descriptive phrase invoking attributes of the object (eg: *"the blue mug"*), rather than having a unique name for each object. The set of such objects in these environments keeps changing, and sometimes even their properties might (eg: a water bottle becomes lighter as it gets emptied). This makes it near impossible to catalog the objects present, and their attributes, requiring robots to use perception to ground such descriptions of objects. Further, it is impossible to determine beforehand the attributes that people are likely to use such objects, and collect annotations for them. Thus to learn perceptual models for objects and attributes, a robot needs to be able to acquire labeled examples during interactions with users. In this work, we introduce the framework of opportunistic active learning, where a robot queries for labeled examples that are not immediately required, in anticipation of using them for future interactions. Full details are available in Thomason et al. (2017).

4.1. Opportunistic Active Learning

Active learning identifies data points from a pool of unlabeled examples whose labels, if made available, are most likely to improve the predictions of a supervised model. Opportunistic Active Learning (OAL) is a setting that incorporates active learning queries into interactive tasks. Let $O = \{o_1, o_2, \ldots o_n\}$ be a set of examples, and $M = \{m_1, m_2, \ldots m_k\}$ be supervised models trained for different concepts, using these examples. For the problem of understanding natural-language object descriptions, O corresponds to the set of objects, M corresponds to the set of possible concepts that can be used to describe the objects, for example their categories (such as *ball* or *bottle*) or perceptual properties (such as *red* or *tall*).

In each interaction, an agent is presented with some subset $O_A \subseteq O$, and must make a decision based on some subset of the models $M_A \subseteq M$. Given a set of candidate objects O_A and a natural language description l, M_A would be the set of classifiers corresponding to perceptual predicates present in l. The decision made by the agent is a guess about which object is being described by l. The agent receives a score or reward based on this decision, and needs to maximize expected reward across a series of such interactions. In the task of object retrieval, this is a 0/1 value indicating whether the guess was correct, and the agent needs to maximize the average guess success rate.

During the interaction, the agent may also query for the label of any of the examples present in the interaction $o \in O_A$, for any model $m \in M$. The agent is said to be opportunistic when it chooses to query for a label $m \notin M_A$, as this label will not affect the decision made in the current interaction, and can only help with future interactions. For example, given a description "*the red box*", asking whether an object is *red*, could help the agent make a better guess, but asking whether an object is *round*, would be an opportunistic query. Queries have a cost, and hence the agent needs to trade-off the number of queries with the success at guessing across interactions.

The agent participates in a sequence of such interactions, and the models improve from labels acquired over multiple interactions. Thus the agent's expected reward per interaction is expected to improve as more interactions are completed.

This setting differs from the traditional application of active learning in the following key ways:

- The agent cannot query for the label of any example from the unlabeled pool. It is restricted to the set of objects available in the current interaction, O_A.
- The agent is evaluated on the reward per interaction, rather than the final accuracy of the models in M.
- The agent may make opportunistic queries (for models $m \notin M_A$) that are not relevant to the current task.

Due to these differences, this setting provides challenges not seen in most active learning scenarios:

- Since the agent never sees the entire pool of unlabeled examples, it can neither choose queries that are globally optimal, nor use variance reduction strategies that still use near-optimal queries (such as sampling from a beam of near globally optimal queries).
- Since the agent is evaluated on task completion, it must learn to trade-off finishing the task with querying to improve the models.
- The agent needs to estimate the usefulness of a model across multiple interactions, to identify good opportunistic queries.

4.2. Object Retrieval Task

To test the effectiveness of opportunistic active learning, we created an object identification task using a real robot. Figure 4.1 shows the physical setup of our task.

We split the set of objects in the current interaction O_A into an active training set O_A^{tr}, and an active test set O_A^{te}. The target object being described is in the active test set and the robot can query objects present in the active training set. This ensures that the robot needs to learn generalizable perceptual classifiers. It also simulates the situation where the target object is in a different room, and the robot needs to query about local objects (active training set) to learn classifiers that can be used later to identify the target.

The human participant and robot both started facing Table 2, which held the active test set. The tables flanking the robot (Tables 1 and 3) contained objects in the active training set.

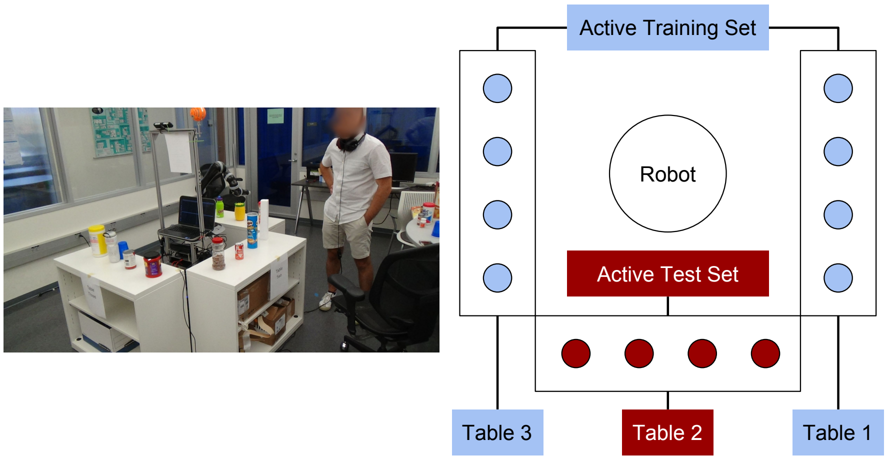


Figure 4.1: Participants described an object on Table 2 from the *active test set* to the robot in natural language, then answered the robot's questions about the objects in its *active training set* on the side Tables 1 and 3 before the robot guessed the described target object.

Human participants engaged in a dialog with the robot.[1] The robot asked the human to describe one of the four objects in its active test set with a noun phrase. Participants were primed to describe objects with properties, rather than categories, given the motivating example "a fuzzy black rectangle" for "an eraser." They were also told that the robot had both looked at and interacted with the objects physically using its arm.

Natural Language Grounding. To connect the noun phrases offered by participants to sensory perception, the robot stripped stopwords from the phrase and considered all remaining words as perceptual predicates. For each perceptual predicate, the robot trained an SVM classifier based on multimodal features. See Thomason et al. (2017) for details. We did not restrict the choice of words that participants were allowed to use to describe objects, so our system learned from an open vocabulary. For every predicate $p \in P$ for P the set of predicates known to the agent and object $o \in O_A$, a decision $d(p, o) \in \{-1, 1\}$ and a confidence [2] $\kappa(p, o)$ in that decision are calculated.

Active Learning Dialog Policy. After the participant described a chosen target object in natural language, the robot asked a fixed number of questions about objects in its active training set before guessing a target object. The robot chose between two types of questions -

- Label queries - A yes/no question about whether a predicate p applies to a specific object $o \in O_A^{tr}$, e.g. "Is this object yellow?".
- Example queries - Asking for an object $o \in O_A^{tr}$, that can be described by a predicate p, e.g. "Show me a white object in this set.". This is used for acquiring positive examples since most predicates tend to be sparse. [3]

[1] View a demonstration video of the robot system and dialog agents here: https://youtu.be/f-CnIF92_wo

[2] Cohen's kappa estimated from cross-validation performance on available examples.

[3] Alternately, we could ask for all positive examples for the predicate in the active training set, but we chose to query for a single example to allow the agent to minimize the amount of supervision obtained

In a label query, to select the predicate p and object $o \in O_A^{tr}$ to ask about, we first find the objects in O_A^{tr} with the lowest confidence κ per predicate (ties broken randomly).

$$o_{\min}(p) = \text{argmin}_{o \in O_A^{tr}}(\kappa(p, o)).$$

and sample predicates inversely proportional to their confidence in their least confident labels.

$$prob(p) = \frac{1 - \kappa(p, o_{\min}(p))}{\sum_{q \in P \setminus \{p\}} 1 - \kappa(q, o_{\min}(q))}. \tag{4.1}$$

For example queries, a predicate p was selected uniformly at random from those with insufficient data to fit a classifier.

The robot updated relevant perceptual classifiers with each answer, and after all questions, identified the best guess $o^* \in O_A^{te}$, using classifiers of predicates $P_A \subseteq P$ present in the description as follows,

$$o^* = \text{argmax}_{o \in O_A^{te}} \left(\sum_{p \in P_A} d(p, o)\kappa(p, o) \right). \tag{4.2}$$

If the robot guessed incorrectly, the human pointed out the correct object. The target object was then considered a positive example for predicates P_A.

4.3. Experiments

We used a dataset of 32 objects (Figure 4.2) explored in Sinapov et al. (2016), divided in 4 folds of 8 objects each. The folds were indexed {0, 1, 2, 3}.

Two dialog agents controlling the robot were compared. The *baseline* agent was only allowed to ask questions about the predicates relevant to the current dialog. That is, if a person described the target object as "a blue cylinder," then the *baseline* agent could only ask about "blue" and "cylinder." By contrast, the *inquisitive* agent was allowed to ask questions about any predicate it had previously heard. Thus, the *inquisitive* agent could ask about "heavy" even if a user used "a blue cylinder" to describe the target object. The *inquisitive* agent also asked 2 extra questions per dialog, making it both more talkative and less task-oriented than the *baseline* agent. At test time, both agents behaved like the *baseline* for fair comparison.

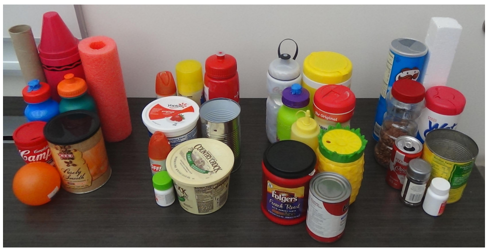


Figure 4.2: The objects used in our experiments, from fold 0 on the far left to fold 3 on the far right.

The agents were tested across three **rounds**, with objects from the active test set moving to the active training set between rounds. Objects in the active test set were always novel to the robot. Between rounds, the dialogs that the agents had with their participants were aggregated and new predicate classifiers were trained to use in the next round.

We hypothesized that:

1. The *inquisitive* agent would guess the correct object more often than the *baseline* agent.

2. Users would not qualitatively dislike the *inquisitive* agent for asking too many questions and being off-topic compared to the *baseline* agent.

4.4. Experimental Results

Five participants played two games each with the robot for each agent in each round, and filled an exit survey afterwards.

Table 4.1 shows the robot's average correctness across rounds between the two agents. The *inquisitive* agent consistently outperforms the *baseline* agent at identifying the correct object, including round 3 where they use the same policy. Since there is a considerable overlap of predicates across rounds, the opportunistic strategy of the *inquisitive* agent is beneficial.

Round	Baseline	Inquisitive
1	0.175	0.35
2	0.225	0.325
3	0.175	0.325

Table 4.1: Fraction of successful guesses made in each condition per round

Exit surveys show that the *inquisitive* agent is perceived as asking too many questions, only when they are off topic, despite the fact that it always asks more questions than the *baseline* agent in the first 2 rounds. However, it was rated as being more fun and usable, presumably because of its higher success at guessing objects correctly.

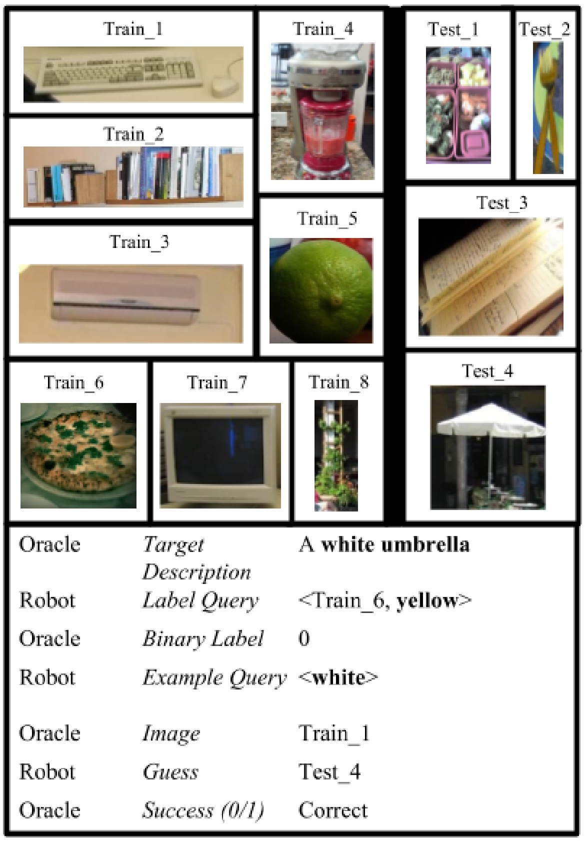
Chapter 5

Learning a Policy for Opportunistic Active Learning

In the previous work (chapter 4), we demonstrated that an interactive robot following an opportunistic active learning policy is better able to ground natural language descriptions of objects across interactions. However, in that work, we compared two static dialog policies that the robot could use for the task. In this work, we learn a dialog policy from interactions using Reinforcement Learning. Full details are available in Padmakumar et al. (2018).

5.1. Task Setup

We consider the same task as in the previous work (section 4.2). However, we set it up in simulation using the Visual Genome dataset (Krishna et al., 2017) as we need a large number of dialogs to learn a dialog policy. The Visual Genome dataset contains images with regions (crops) annotated with natural-language descriptions. Bounding boxes of objects present in the image are also annotated, along with attributes of objects. Region descriptions, objects and attributes are annotated using unrestricted natural language, which leads to a diverse set of predicates. Using the annotations, we can associate a list of objects and attributes relevant to each image region, and use these to answer queries from the agent.



Oracle	*Target Description*	**A white umbrella**
Robot	*Label Query*	<Train_6, **yellow**>
Oracle	*Binary Label*	0
Robot	*Example Query*	<**white**>
Oracle	*Image*	Train_1
Robot	*Guess*	Test_4
Oracle	*Success (0/1)*	Correct

Figure 5.1: A sample OAL interaction. Perceptual predicates are marked in bold.

A sample interaction is seen in figure 5.1. For each interaction, we uniformly sample

4 regions to form the active test set, and 8 regions to form the active training set. [1] One region is then uniformly sampled from the active test set to be the target object. Its description, from annotations, is provided to the agent to be grounded. Following this, the agent can ask label and example queries on the active training set, before guessing which object was being described. The objects and attributes associated with active training regions are used to answer queries. A predicate is labeled as being applicable to a region if it is present in the list of objects and attributes associated with the region.

5.2. Methodology

We extract predicates from descriptions as in previous work (section 4.2). Predicates are grounded using binary SVMs trained over deep features extracted from images. These are obtained from the penultimate layer of the VGG network (Simonyan and Zisserman, 2014a) pretrained on ImageNet (Russakovsky et al., 2015).

5.2.1. Grounding Descriptions

Grounding is performed similar to previous work (section 4.2). Given predicates $P_A \subseteq P$ present in the target description l, a decision $d(p, o) \in \{-1, 1\}$ from the classifier for predicate p for object o, and the confidence of the classifier $C(p)$ (estimated F1 from cross-validation on acquired labels), the best guess o^* is computed as,

$$o^* = \text{argmax}_{o \in O_A^{te}} \sum_{p \in P_A} d(p, o) * C(p)$$

5.2.2. MDP Formulation

We model interactions as episodes in a Markov Decision Process (MDP) (section 2.3.1). At any point, the agent is in a state consisting of the VGG features of the regions in the current interaction, the predicates in the current description, and the agent's classifiers. The agent can choose from the set of actions which includes an action for guessing, and an action for each possible query the agent can currently make, including both label and example queries. The guess action always terminates the episode, and query actions transition the agent to a new state as one of the classifiers gets updated. The agent gets a reward for each action taken. Query actions have a small negative reward, and guessing results is a large positive reward when the guess is correct, and a large negative reward when the guess is incorrect. In our experiments, we treat the reward values as hyperparameters that can be tuned. The best results were obtained with a reward of 200 for a correct guess, -100 for an incorrect guess and -1 for each query.

5.2.3. Identifying Candidate Queries

Ideally, we would like the agent to learn a policy over all possible queries. However, as the number of predicates the agent knows continuously increases as it completes interactions, processing the full action space quickly becomes intractable. Hence we sample

[1] The regions in the dataset are divided into separate pools from which the active training and active test sets are sampled (described as classifier-training and classifier-test sets in section 5.3), to ensure that the agent needs to learn classifiers that generalize across objects.

a few promising queries and learn a policy to choose between them. In our previous work, predicates were sampled according to a distribution that weighted them inversely proportional to the confidence in their current classifiers. However, if the space of possible predicates is large, then this results in no classifier obtaining a reasonable number of training examples. In this scenario, it is desirable to focus on a small number of predicates, possibly stopping the improvement on a predicate once the classifier for it has been sufficiently improved. We sample queries from a distribution designed to capture this intuition. The probability assigned to a predicate by this distribution increases linearly, for confidence below a threshold, and decreases linearly thereafter.

5.2.4. Policy Learning

We use the REINFORCE algorithm (section 2.3.3) to learn a policy for the MDP. The state consists of the predicates in the current description, the candidate objects, and the current classifiers. Since both the number of candidate objects and classifiers varies, and the latter is quite large, it is necessary to identify useful features for the task to obtain a vector representation needed by most learning algorithms. In our problem setting, the number of candidate actions available to the agent in a given state is variable. Hence we need to create features for state-action pairs, rather than just states.

The object retrieval task consists of two parts – identifying useful queries to improve classifiers, and correctly guessing the image being referred to by a given description. The current dialog length is also provided to influence the trade-off between guessing and querying.

Guess-success features

- Lowest, highest, second highest, and average confidence among classifiers of predicates in the description – learned thresholds on these values can be useful to decide whether to trust the guess.
- Highest score among regions in the active test set, and the differences between this and the second highest, and average scores respectively – a good guess is expected to have a high score to indicate relevance to the description, and substantial differences would indicate that the guess is discriminative. Similar features are also formed using the unweighted sum of decisions.
- An indicator of whether the two most confident classifiers agree on the decision of the top scoring region, which increases the likelihood of its being correct.

Query-evaluation features

- Indicator of whether the predicate is new or already has a classifier – this allows the policy to decide between strengthening existing classifiers or creating classifiers for novel predicates.
- Current confidence of the classifier for the predicate – as there is more to be gained from improving a poor classifier.
- Fraction of previous dialogs in which the predicate has been used, and the agent's success rate in these – as there is more to be gained from improving a frequently used predicate but less if the agent already makes enough correct guesses for it.

- Is the query opportunistic – as these will not help the current guess.

Label queries also have an image region specified, and for these we have additional features that use the VGG feature space in which the region is represented for classification:

- Margin of the image region from the hyperplane of the classifier of the predicate – motivated by uncertainty sampling.
- Average cosine distance of the image region to others in the dataset – motivated by density weighting to avoid outliers.
- Fraction of the k-nearest neighbors of the region that are unlabeled for this predicate – motivated by density weighting to identify a data point that can influence many labels.

5.3. Experiments

Sampling Dialogs

We want the agent to learn a policy that is independent of the actual predicates present at policy training and policy test time. In order to be able to evaluate this, we divide the set of possible regions in the Visual Genome dataset into policy training and policy test regions as follows. We select all objects and attributes present in at least 1,000 regions. Half of these were randomly assigned to the policy test set. All regions that contain one of these objects or attributes are assigned to the policy test set, and the rest to the policy training set. Thus regions seen at test time may contain predicates seen during training, but will definitely contain at least one novel predicate. Further, the policy training and policy test sets are respectively partitioned into a classifier training and classifier test set using a uniform 60-40 split.

During policy training, the active training set of each dialog is sampled from the classifier-training subset of the policy-training regions, and the active test set of the dialog is sampled from the classifier-test subset of the policy-training set. During policy testing, the active training set of each dialog is sampled from the classifier training subset of the policy test regions, and the active test set of the dialog is sampled from the classifier test subset of the policy test set.

5.3.1. Experiment phases

Our baseline is a static policy similar to that used in previous work (4.2).

For efficiency, we run dialogs in batches, and perform classifier and policy updates at the end of each batch. We use batches of 100 dialogs each. Our experiment runs in 3 phases:

- Initialization – Since learning starting with a random policy can be difficult, we first run batches of dialogs on the policy training set using the static policy, and update the RL policy from these episodes. This “supervised” learning phase is used to initialize the RL policy.

- Training – We run batches of dialogs on the policy training set using the RL policy, starting it without any classifiers. In this phase, the policy is updated using its own experience.
- Testing – We fix the parameters of the RL policy, and run batches of dialogs on the policy test set. During this phase, the agent is again reset to start with no classifiers. We do this to ensure that performance improvements seen at test time are purely from learning a strategy for opportunistic active learning, not from acquiring useful classifiers in the process of learning the policy.

5.4. Experimental Results and Analysis

We initialize the policy with 10 batches of dialogs, and then train on another 10 batches of dialogs, both sampled from the policy training set. Following this, the policy weights are fixed, the agent is reset to start with no classifiers, and we test on 10 batches of dialogs from the policy test set. Table 5.1 compares the average success rate (fraction of successful dialogs in which the correct object is identified), and average dialog length (average number of system turns) of the best learned policy, and the baseline static policy on the final batch of testing. We also compare the effect of ablating the two main groups of features. The learned agent guesses correctly in a significantly higher fraction of dialogs compared to the static agent, using a significantly lower number of questions per dialog.

Policy	Success rate	Average Dialog Length
Learned	**0.44**	**12.95**
–Guess	0.37	**6.12**
–Query	0.35	**6.16**
Static	0.29	16

Table 5.1: Results on dialogs sampled from the policy test set after 10 batches of classifier training. *–Guess* and *–Query* are conditions with the guess and query features, respectively, ablated. Boldface indicates that the difference in that metric with respect to the *Static* policy is statistically significant according to an unpaired Welch t-test with $p < 0.05$.

We also explored ablating individual features. We found that the effect of ablating most single features is similar to that of ablating a group of features. The mean success rate decreases compared to the full policy with all features. It remains better than that of the static policy, but in most cases the difference stops being statistically significant. An interesting result is that removal of features involving the predictions of the second best classifier has more effect than that of the best classifier. This is possibly because when noisy classifiers are in use, support of multiple classifiers is helpful.

Qualitatively, we found that the dialog success rate was higher for both short, and very long dialogs, with a decrease for dialogs of intermediate length. This suggests that longer dialogs are used to accumulate labels via opportunistic off-topic questions, as opposed to on-topic questions. The learned policy still suffers from high variance in dialog length suggesting that trading off task completion against model improvement is a difficult decision to learn. We find that the labels collected by the learned policy are more equitably distributed across predicates than the static policy, resulting in a tendency to have fewer classifiers of low confidence. This suggests that the policy learns to focus on a few predicates, as the baseline does, but learn all of these equally well, in contrast to the baseline which has much higher variance in the number of labels collected per predicate.

Chapter 6

Proposed Work

6.1. Learning to Ground Natural Language Object Descriptions Using Joint Embeddings

In our previous work (chapters 4 and 5), we have used binary classifiers for grounding perceptual predicates. For processing images, we use features extracted from the VGG network (Simonyan and Zisserman, 2014b) to leverage the recent success of deep neural networks in image classification. Learning a multi-class classifier allows the model to learn correlations between labels, but this requires the set of labels to be pre-defined. This is typically not desirable when trying to understand unrestricted natural language descriptions, as a wide variety of concepts, and words may be used to describe objects. However, learning binary classifiers prevents us from exploiting similarities between labels not present in ImageNet (and hence not seen during the pretraining as a multi-class classifier).

For example, "*red*" and "*scarlet*" are very similar colors, "*red*" and "*blue*" are both colors but less similar in meaning otherwise, "*red*" and "*apple*" are not similar in meaning but many apples are red. Word embeddings have been shown to capture many such similarity and relatedness properties (Mikolov et al., 2013). These are shown to be enhanced by training multimodal vectors (Lazaridou et al., 2015b). There are also works

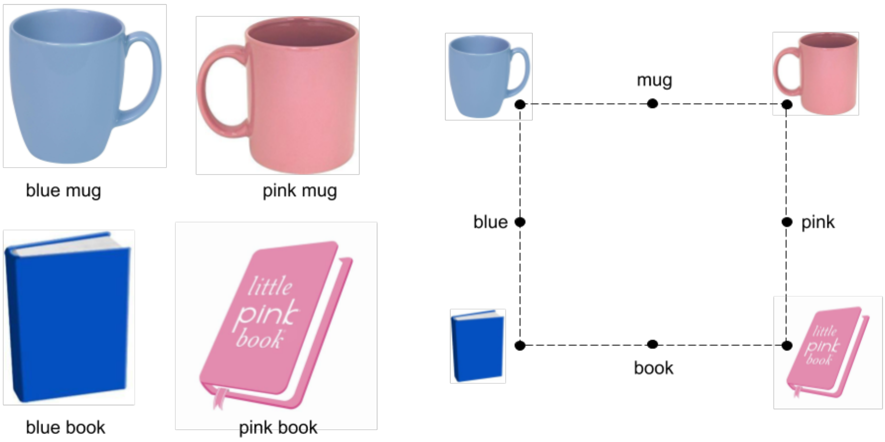


Figure 6.1: Expected joint space for a simple dataset of 4 objects.

on learning networks to score how well a natural language description or caption applies to an image (Hu et al., 2016; Xiao et al., 2017; Wang et al., 2016), some of which test the effectiveness of these methods for retrieving images based on captions - the problem setting we use for grounding (Hu et al., 2016). However these typically use embeddings of the entire sentence or phrase, instead of grounding representations of individual words and combining them.

We wish to design a model for grounding that compares vector representations of images with word (as opposed to phrase/ sentence) embeddings. We expect these to be able to learn from fewer training examples, as words recur more frequently across descriptions. We also expect these to generalize better to novel descriptions, or to different datasets, as they can be useful if the learned representations of some words transfer.

We propose to do this by projecting pre-trained vector representations of the images and words in their descriptions to a learned joint space, such that projected word vectors are close to projected images they are related to, and far from projected images they are not related to. A simple example with 4 images, with Euclidean distance as the distance metric, is in figure 6.1. The sum of distances from the projected image of the blue mug to the words blue and mug, is less than this sum for any other image. The same holds for all descriptions.

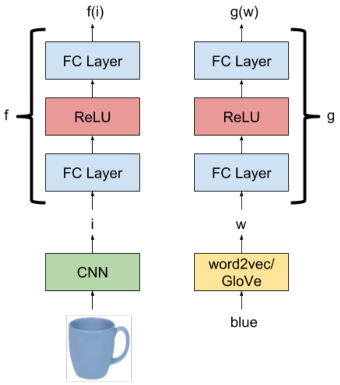


Figure 6.2: Networks to embed and project images and words.

More formally, given an image representation i, and its language description $l = (w_1, w_2, \dots w_k)$, and a distance function d, we learn projection functions f and g such that for any other image i',

$$\begin{aligned} & d(f(i), g(w_j)) \leq d(f(i'), g(w_j)) \\ & \forall\, j \in \{1, 2, \dots k\} \end{aligned} \tag{6.1}$$

We will represent the functions f and g using neural networks with a single hidden layer (figure 6.2), and learn their parameters using a ranking loss to capture the above constraints (Wang et al., 2016).

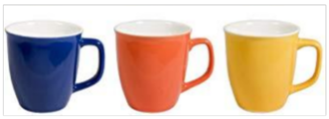


Figure 6.3: Using the space in figure 6.1 to ground “dark blue mug” among the these objects, we can probably identify the correct object without understanding the word “dark”.

To ground a novel description $l' = (w'_1, w'_2, \dots w'_{k'})$, we find the image i_{min} which when projected minimizes the average distance to the projected words in the description,

$$i_{min} = argmin_i \left(\frac{1}{k'} \Sigma_{j=1}^{k'} d(f(i), g(w'_j)) \right) \tag{6.2}$$

A similar model is that of Wang et al. (2016) who project embeddings of images, and embeddings of their corresponding descriptive phrases into a joint space, using projections and a ranking loss. We expect the projecting word embeddings would result in better performance because it is possible to directly exploit words, which can be grounded visually, that recur across descriptions. Such a model would be more robust to the presence of unseen words, as it may be possible to ground a description based on a subset of the attributes provided. For example, suppose the space learned in figure 6.1 has to be used to ground the phrase “dark blue mug" to one of the objects present in 6.3. Since the target mug can be identified using the attributes "blue" and "mug", it is reasonable to expect that the right object would be identified. However, if the entire phrase has to be embedded before projection, it is likely that the addition of the extra attribute "dark" will produce an input vector very different from those seen at train time, and may not be projected in a manner that its meaning can be grounded well.

We intend to compare the performance of our model with some competitive baselines (Guadarrama et al., 2016; Hu et al., 2016; Xiao et al., 2017; Wang et al., 2016) on existing datasets of images with language descriptions (Hu et al., 2016). This would be an additional contribution as these different types of grounding methods have not previously been compared in a single dataset.

6.2. Identifying Useful Clarification Questions for Grounding Object Descriptions

In our previous work on clarification of natural language commands (chapter 3), we allow the robot to confirm or request for specific semantic roles in a command. If the command involves manipulating an object, this allows the robot to ask the user for a description of the object, or ask whether a specific object is the one to be manipulated. If the robot does not fully understand the object description received, the above clarification questions are tedious to resolve the ambiguity, and become impractical if the set of candidate objects is large. Humans typically use categories and attributes to describe objects, and if appropriately chosen, a robot can use these to ask clarifications.

An example is shown in table 6.1, where the robot tries to find other attributes, such as "*white*", that can be used to describe the target object that could not be determined using the original description "small china mug".

Table 6.1: An sample clarification dialog

ROBOT	How can I help?
USER	Bring me the small china mug.
ROBOT	What should I bring?
USER	The small china mug
ROBOT	Is that object white?
USER	yes

We would ideally like the robot to choose question that is most likely to decrease the size of the search space. That is, it is expected to provide the most information. When using binary classifiers for grounding, this can be estimated using the entropy of the set of candidates, based on each classifier's predictions. Let $p^{(c)}$ be the probability that the attribute predicted by classifier c is true for an object in the set of candidates. This can be calculated as the fraction of objects in the candidate set for which the classifier predicts *true*. Then the entropy $E^{(c)}$ of the candidate set based on this attribute is calculated as follows

$$E^{(c)} = -p^{(c)} \log p^{(c)} - (1 - p^{(c)}) \log \left(1 - p^{(c)}\right) \tag{6.3}$$

A higher entropy indicates a higher information content. Hence the best question can be chosen as the one corresponding to the classifier of maximum entropy.

$$c^* = argmax_c E^{(c)} \tag{6.4}$$

With binary classifiers, this chooses an attribute such that whether this is true or false for the target, the search space is reduced by a significant amount. For example, if half the objects the robot is considering are white, asking whether the target is white is helpful. If only one object is red, asking asking whether the target is red is less helpful, because if the user gives a negative reply, most of the objects still remain as candidates. In this example, the configuration the entropy for white is much higher than the entropy for red.

If we use joint embeddings for grounding, adopting the same principle is less trivial, because it is unclear how to compute the probability $p^{(c)}$ for a concept c. The following are a few ways to estimate this –

- Learn a distance threshold t, such that given projections of an image $f(i)$ and a word $g(w)$, w is applicable for image i if the distance between them, $d(f(i), g(w)) \leq t$. Then $p^{(c)}$ is the fraction of images that are closer than this threshold, that is, $d(f(i), g(w)) \leq t$.

- Cluster the images in the projected space. Any word w whose projection lies within a cluster is applicable to all images in the cluster. Then $p^{(c)}$ is the fraction of images that lie in the cluster within which c is projected to.

It is possible that choosing this method may be dependent on the nature of the learned joint space. It may also be possible to cast the problem as an optimization problem without using heuristics. This requires further exploration.

Related to this is work on learning end-to-end neural networks on object guessing games (Das et al., 2017b; De Vries et al., 2017). However, in these works, since both agents are trained simultaneously, there is no constraint besides pretraining, that requires them to learn the same meanings for words that humans would use. Also related is work on learning to ask discriminative questions (Li et al., 2017). But this only considers differentiating between two images. Further, the questions are not related to any prior understanding by the system, whereas, we intend to identify questions that are discriminate between images that the system believes are candidates to satisfy a given description.

6.3. Learning a Policy for Clarification Questions using Uncertain Models

We have proposed methods to identify useful clarification questions for grounding natural language descriptions of objects. However, these are based on information from models that could be uncertain. For instance, we propose that the robot choose a question that maximizes entropy. However, the entropy is computed based on predictions made by uncertain classifiers. If a particularly poor classifier is chosen, which makes an erroneous prediction on the target object, the robot may either falsely discard it, based on the response to the clarification, or may not be able to give sufficient importance to the question and prune the search space as expected.

For distributional models, the questions are chosen on even weaker heuristics that we expect would model entropy. It may be desirable to learn a policy that selects a question from a beam of candidates proposed by such heuristic methods, instead of relying on them completely. This can be learned using reinforcement learning over simulated dialogs similar to that used in our previous work (chapter 5).

An important challenge in this setting is identifying features that can be used to determine confidence in distance measures in a learned embedding space. A possible heuristic for this could be to the fraction of positive examples within a distance threshold (a measure of cluster purity).

Another challenge with clarification dialogs in distributional spaces is how to combine the information from the original description with that obtained from clarification. We could treat all predicates equally and average them (equation 6.2). But it is possible that some computed distances are more reliable than others, or some words may be more salient in a description (for example, if there is only one mug in the set of candidate objects, additional attributes are superfluous in identifying it).

When classifiers are used for grounding, it is possible handcraft a good rule to combine predictions of the classifiers, and confidences to identify the target object (equation 5.2.1). However, if we instead have to use distances in a learned space, it may be beneficial to learn how to combine distances between projected images and words. This is non-trivial because the number of words involved changes over time, and the number of images under consideration could be variable. Some possible models for this include -

- A neural network that maps the current score of an object, its distance to a new word, and whether this word applies to the target, to a new score. This would initially be applied sequentially to words in the description, and then to subsequent new words for each question-answer pair. After each update, scores would have to be normalized.

- Fix an upper bound on the number of predicates, m, and learn a network to map the current score of an object, distances to up to m words, with a default value when fewer predicates are present, and whether they apply to the target, to an updated score. The network can potentially use representations of the words themselves to learn correlations between their meanings, and the final score. However, this requires discarding some words when too many are present, and also requires a separate normalization step.

- Fix upper bounds both on the number of objects and number of words, and use features of the objects and words, the current scores of all objects, and distances and relevances of all words to all objects. This makes no independence assumptions but uses a very large input space.

Further, if we have a measure of confidence on the distance estimate, that would also be a relevant feature, and could potentially result in better beliefs.

Chapter 7

Bonus Contributions

This chapter covers areas of future work that augment the proposed work, that may be included in the final thesis. This involves extending the proposed model for perceptual grounding based on joint embeddings to

- Incorporate linguistic and visual context.
- Handle multimodal representations of objects, for example using audio or haptic information.

7.1. Context Sensitive Joint Embeddings for Language Grounding

Context is important for understanding language, and often difficult to capture. An example where linguistic context affects the meaning of a word is distinguishing word senses:

> *Swing the baseball* ***bat****.*
> *Don't touch the dead* ***bat****.*

Visual context may also affect the use and meaning of a word, for example whether relative adjectives such as *big* and *small* apply. An example can be seen in figure 7.1.

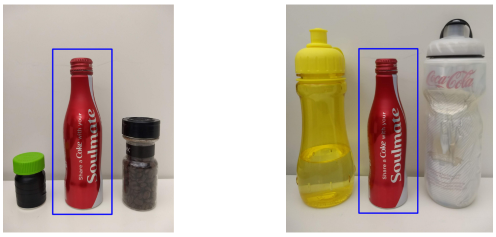


Figure 7.1: The highlighted bottle is the same in both images but can be described as *the big bottle* in the image on the left and *the small bottle* in the image on the right.

Context sensitive word embeddings have recently been shown to be more effective in a number of tasks including question answering, textual entailment, semantic role labeling, coreference resolution, named entity extraction, and sentiment analysis (Peters et al., 2018). There is also some prior work on using different forms of visual context in understanding object descriptions (Hu et al., 2016; Misra et al., 2017b).

An interesting extension of our proposed method for grounding object descriptions (section 6.1) would be to explore the effect of using context sensitive representations of words and images in that method. For word representations, we will use

ELMo embeddings (Peters et al., 2018) mentioned above, and for contextual visual representations, the following are two possibilities:

- A representation learned using an embedding of the image using a convolutional neural network (CNN), an embedding of the bounding box of the object using a CNN, and relative coordinates of the bounding box (Hu et al., 2016).
- Obtain CNN embeddings of object proposals from an object detection network such as R-CNN (Ren et al., 2015), and combine them using an attention network (Anderson et al., 2018).

7.2. Multimodal Grounding of Object Descriptions Using Joint Embeddings

Robots can sense the world through modalities other than vision, for example sound through a microphone or by manipulating objects with an arm. Prior work has shown that when people are allowed to handle objects before describing them, they may use non-visual predicates such as *heavy* or *rattling* to describe them. Using multimodal features has been shown to allow a robot to better ground object descriptions, and are essential when non-visual predicates are used (Thomason et al., 2016).

It would be interesting to extend our proposal on using joint embeddings for grounding object descriptions (section 6.1) to incorporate more modalities. A challenge here is that collecting multimodal data using a robot is time consuming, thus requiring our methods to be sample efficient. Also, ideally the method should be usable for objects for which only visual features are available, so that we can leverage the availability of paired language and vision datasets.

Given an object o with features in modalities $m \in \mathbb{M}^{(o)}$, let $o^{(m)}$ be the feature representation of o in modality m. Also, let $w_1, w_2, \dots w_k$ be embeddings of the words in the description l. We use a projection function g to project words, and projection functions $f^{(m)}$ to project feature representations in modality m, $o^{(m)}$, to the joint space. Then, the following two methods can be used to measure the distance $D(o, l)$ between an object o and a description l, given a distance function d (such as cosine distance) in the space.

- Average distance to all modalities:

$$D(o,l) = \frac{1}{\mathbb{M}^{(o)}} \frac{1}{k} \sum_{m \in \mathbb{M}^{(o)}} \sum_{i=1}^{k} d(f^{(m)}(o^{(m)}), g(w_i)) \tag{7.1}$$

- Averaging the closest modality to each word

$$D(o,l) = \frac{1}{k} \sum_{i=1}^{k} \min_{m \in \mathbb{M}^{(o)}} d(f^{(m)}(o^{(m)}), g(w_i)) \tag{7.2}$$

  This metric is expected to account for the fact that a modality may only be able to capture the meanings of some words, as a word needs to be close to an object only in the relevant modality.

The above metrics allow for training the projection functions g and $f^{(m)}$ using objects with annotated descriptions, that do not necessarily have feature representations in all modalities. The learned representations would be expected to be better for those modalities in which features of more objects are present. They would be trained using a ranking loss that enforces the constraint that given an object o with its paired description l

$$D(o, l) \leq D(o', l) \; \forall \; o' \neq o \tag{7.3}$$

$$D(o, l) \leq D(o, l') \; \forall \; l' \neq l \tag{7.4}$$

Chapter 8

Conclusion

Facilitating natural language communication between humans and robots faces a number of challenges, including the need to ground language in perception, the ability to adapt to changes in the environment and novel uses of language, and to deal with uncertainty on the part of the robot. Robots can use two types of questions to achieve this - active learning queries to elicit knowledge about the environment that can be used to improve perceptual models, and clarification questions that confirm he robot's hypotheses, or elicit specific information required to complete a task. The robot should also be able to learn how to conduct such dialogs through interaction – which can be achieved by dialog policy learning. We present completed work on jointly improving semantic parsers from and learning a dialog policy for clarification dialogs, that improve a robot's ability to understand natural language commands. We introduce the framework of opportunistic active learning, where a robot introduces opportunistic queries, that may not be immediately relevant, into an interaction in the hope of improving performance in future interactions. We demonstrate the usefulness of this framework in learning to ground natural language descriptions of objects, and learn a dialog policy for such interactions. We propose a new model for grounding natural language descriptions of objects based on joint embeddings, and propose to conduct a systematic comparison of different types of perceptual grounding models. We also suggest possible extensions of this model to incorporate context and multimodal object representations. We also propose a method to identify useful clarification questions when trying to understand natural language descriptions of objects, and propose to learn a dialog system that makes use of these.

Bibliography

Anderson, P., X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang
2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, P. 6.

Artzi, Y. and L. Zettlemoyer
2011. Bootstrapping Semantic Parsers from Conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Pp. 421–432.

Arumugam, D., S. Karamcheti, N. Gopalan, E. C. Williams, M. Rhee, L. L. Wong, and S. Tellex
2018. Grounding natural language instructions to semantic goal representations for abstraction and generalization. *Autonomous Robots*.

Bachman, P., A. Sordoni, and A. Trischler
2017. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, eds., volume 70, Pp. 301–310, Sydney, Australia. PMLR.

Bastianelli, E., D. Croce, A. Vanzo, R. Basili, and D. Nardi
2016. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, Pp. 2747–2753. AAAI Press.

Berant, J., A. Chou, R. Frostig, and P. Liang
2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bisk, Y., D. Yuret, and D. 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*, Pp. 751–761.

Blukis, V., D. Misra, R. A. Knepper, and Y. Artzi
2018. Mapping navigation instructions to continuous control actions with position visitation prediction. In *Proceedings of the Conference on Robot Learning*.

Bordes, A. and J. Weston
2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Brinker, K.
2006. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, Pp. 206–213. Springer-Verlag.

Chang, A., A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang
2017. Matterport3d: Learning from rgb-d data in indoor environments. In *3D Vision*.

Chaurasia, S. and R. J. Mooney
2017. Dialog for language to code. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP-17)*, Pp. 175–180, Taipei, Taiwan.

Chen, D. L. and R. J. Mooney
2011. Learning to interpret natural language navigation instructions from observations. In *AAAI*.

Das, A., S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra
2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, A., S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra
2017a. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, Pp. 326 – 335.

Das, A., S. Kottur, J. M. Moura, S. Lee, and D. Batra
2017b. Learning cooperative visual dialog agents with deep reinforcement learning. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, Pp. 2970–2979. IEEE.

Daubigney, L., M. Geist, S. Chandramohan, and O. Pietquin
2012. A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimization. *Journal of Selected Topics in Signal Processing*, 6(8):891–902.

De Vries, H., F. Strub, S. Chandar, O. Pietquin, H. Larochelle, and A. Courville
2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dindo, H. and D. Zambuto
2010. A probabilistic approach to learning a visually grounded language model through human-robot interaction. In *International Conference on Intelligent Robots and Systems*, Pp. 760–796, Taipei, Taiwan. IEEE.

Fang, M., Y. Li, and T. Cohn
2017. Learning how to active learn: A deep reinforcement learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL.

Fang, R., M. Doering, and J. Y. Chai
2014. Collaborative models for referring expression generation in situated dialogue. In *AAAI*, Pp. 1544–1550.

Fang, R., C. Liu, L. She, and J. Y. Chai
2013. Towards situated dialogue: Revisiting referring expression generation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Pp. 392–402.

Feng, F., X. Wang, R. Li, and I. Ahmad
2015. Correspondence autoencoders for cross-modal retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(1s):26.

Forbes, M. and Y. Choi
2017. Verb physics: Relative physical knowledge of actions and objects. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 266–276.

Gao, Q., M. Doering, S. Yang, and J. Chai
2016. Physical causality of action verbs in grounded language understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 1814–1824.

Gao, Q., S. Yang, J. Chai, and L. Vanderwende
2018. What action causes this? towards naive physical action-effect prediction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 934–945.

Gašić, M. and S. Young
2014. Gaussian Processes for POMDP-Based Dialogue Manager Optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40.

Geist, M. and O. Pietquin
2010. Kalman Temporal Differences. *Journal of Artificial Intelligence Research*, 39(1):483–532.

Guadarrama, S., E. Rodner, K. Saenko, and T. Darrell
2016. Understanding object descriptions in robotics by open-vocabulary object retrieval and detection. *The International Journal of Robotics Research*, 35(1-3):265–280.

Harnad, S.
1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Hemphill, C. T., J. J. Godfrey, and G. R. Doddington
1990. The atis spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, Pp. 96–101, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hu, H., X. Wu, B. Luo, C. Tao, C. Xu, W. Wu, and Z. Chen
2018. Playing 20 question game with policy-based reinforcement learning. In *EMNLP*.

Hu, R., M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko
2017. Modeling relationships in referential expressions with compositional modular networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, Pp. 4418–4427. IEEE.

Hu, R., H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell
2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 4555–4564.

Kaelbling, L. P., M. L. Littman, and A. R. Cassandra
1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial intelligence.*

Kiddon, C., L. Zettlemoyer, and Y. Choi
2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Pp. 329–339.

Kollar, T., J. Krishnamurthy, and G. Strimel
2013a. Toward interactive grounded language acquisition. In *Robotics: Science and Systems.*

Kollar, T., V. Perera, D. Nardi, and M. Veloso
2013b. Learning Environmental Knowledge from Task-Based Human-Robot Dialog. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, Pp. 4304–4309.

Kottur, S., R. Vedantam, J. M. Moura, and D. Parikh
2016. Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 4985–4994.

Krishna, R., Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei
2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73.

Kulick, J., M. Toussaint, T. Lang, and M. Lopes
2013. Active learning for teaching a robot grounded relational symbols. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, Pp. 1451–1457. AAAI Press.

Kwiatkowski, T., E. Choi, Y. Artzi, and L. Zettlemoyer
2013. Scaling Semantic Parsers with On-the-fly Ontology Matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Lauria, S., G. Bugmann, T. Kyriacou, and E. Klein
2002. Mobile robot programming using natural language. *Robotics and Autonomous Systems*, 38(3-4):171–181.

Lazaridou, A., M. Baroni, et al.
2015a. Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Pp. 153–163.

Lazaridou, A., M. Baroni, et al.
2015b. Combining language and vision with a multimodal skip-gram model. In *ACL*, Pp. 153–163.

Lazaridou, A., E. Bruni, and M. Baroni
2014. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 1403–1414.

Lewis, D. D. and W. A. Gale
1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '94*, Pp. 3–12. Springer London.

Li, X. and Y. Guo
2013. Active learning with multi-label svm classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, Pp. 1479–1485.

Li, X., L. Wang, and E. Sang
. Multi-label SVM active learning for image classification. In *2004 International Conference on Image Processing, 2004. ICIP'04*, volume 4, Pp. 2207–2210. IEEE.

Li, Y., C. Huang, X. Tang, and C. Change Loy
2017. Learning to disambiguate by asking discriminative questions. In *Proceedings of the IEEE International Conference on Computer Vision*, Pp. 3419–3428.

Liu, C., S. Yang, S. Saba-Sadiya, N. Shukla, Y. He, S.-C. Zhu, and J. Chai
2016. Jointly learning grounded task structures from language instruction and visual demonstration. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Pp. 1482–1492.

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

Mei, H., M. Bansal, and M. R. Walter
2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean
2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, Pp. 3111–3119.

Misra, D., J. Langford, and Y. Artzi
2017a. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Pp. 1015–1026. Association for Computational Linguistics.

Misra, I., A. Gupta, and M. Hebert
2017b. From red wine to red tomato: Composition with context. In *CVPR*, volume 2, P. 6.

Mrkšić, N., D. O. Séaghdha, B. Thomson, M. Gašić, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young
2015. Multi-Domain Dialog State Tracking Using Recurrent Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Padmakumar, A., P. Stone, and R. J. Mooney
2018. Learning a policy for opportunistic active learning. In *To Appear in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-18)*, Brussels, Belgium.

Padmakumar, A., J. Thomason, and R. J. Mooney
2017. Integrated learning of dialog strategies and semantic parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Pp. 547–557, Valencia, Spain.

Parde, N., A. Hair, M. Papakostas, K. Tsiakas, M. Dagioglou, V. Karkaletsis, and R. D. Nielsen
2015. Grounding the meaning of words through vision and interactive gameplay. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Pp. 1895–1901, Buenos Aires, Argentina.

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer
2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, Pp. 2227–2237.

Pietquin, O., M. Geist, S. Chandramohan, and H. Frezza-Buet
2011. Sample-efficient batch reinforcement learning for dialogue management optimization. *ACM Transactions on Speech and Language Processing*, 7(3):1–21.

Qi, G.-J., X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang
2009. Two-dimensional multilabel active learning with an efficient online adaptation model for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1880–1897.

Ren, S., K. He, R. Girshick, and J. Sun
2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, Pp. 91–99.

Roy, N. and A. McCallum
2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, Pp. 441–448, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei
2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Serban, I. V., A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau
2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, volume 16, Pp. 3776–3784.

Settles, B.
2010. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Settles, B. and M. Craven
2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP'08*. ACL.

Seung, H. S., M. Opper, and H. Sompolinsky
1992. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, Pp. 287–294, New York, NY, USA. ACM.

Sharma, M. and M. Bilgic
2016. Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, 31(1):164–202.

She, L. and J. Chai
2017. Interactive learning of grounded verb semantics towards human-robot communication. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 1634–1644.

She, L., S. Yang, Y. Cheng, Y. Jia, J. Chai, and N. Xi
2014. Back to the Blocks World: Learning New Actions through Situated Human-Robot Dialogue. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, Pp. 89–97.

Silberer, C. and M. Lapata
2012. Grounded models of semantic representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Pp. 1423–1433. Association for Computational Linguistics.

Silberer, C. and M. Lapata
2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, Pp. 721–732.

Simonyan, K. and A. Zisserman
2014a. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, arXiv:1409.1556.

Simonyan, K. and A. Zisserman
2014b. Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*, abs/1409.1556.

Sinapov, J., P. Khante, M. Svetlik, and P. Stone
2016. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Singh, M., E. Curran, and P. Cunningham
2009. Active learning for multi-label image annotation. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*, Pp. 173–182.

Steedman, M. and J. Baldridge
2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, Pp. 181–224.

Suhr, A., S. Iyer, and Y. Artzi
2018. Learning to map context-dependent sentences to executable formal queries. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Pp. 2238–2249. Association for Computational Linguistics.

Sutton, R. S., A. G. Barto, et al.
1998. *Reinforcement learning: An introduction.* MIT press.

Tellex, S., R. A. Knepper, A. Li, N. Roy, and D. Rus
2014. Asking for Help Using Inverse Semantics. In *Proceedings of the 2016 Robotics: Science and Systems Conference (RSS).*

Theobalt, C., J. Bos, T. Chapman, A. Espinosa-Romero, M. Fraser, G. Hayes, E. Klein, T. Oka, and R. Reeve
2002. Talking to godot: Dialogue with a mobile robot. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 2, Pp. 1338–1343. IEEE.

Thomason, J., A. Padmakumar, J. Sinapov, J. Hart, P. Stone, and R. J. Mooney
2017. Opportunistic active learning for grounding natural language descriptions. Pp. 67–76.

Thomason, J., J. Sinapov, M. Svetlik, P. Stone, and R. Mooney
2016. Learning multi-modal grounded linguistic semantics by playing "I spy". In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, Pp. 3477–3483.

Thomason, J., S. Zhang, R. Mooney, and P. Stone
2015. Learning to Interpret Natural Language Commands through Human-Robot Dialog. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, Pp. 1923–1929.

Thomson, B. and S. Young
2010. Bayesian Update of Dialogue State: A POMDP framework for Spoken Dialogue Systems. *Computer Speech and Language*, 24(4):562–588.

Vasisht, D., A. Damianou, M. Varma, and A. Kapoor
2014. Active learning for sparse bayesian multilabel classification. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD'14*, Pp. 472–481. ACM Press.

Vogel, A., K. Raghunathan, and D. Jurafsky
2010. Eye spy: Improving vision through dialog. In *Association for the Advancement of Artificial Intelligence*, Pp. 175–176.

Wang, L., Y. Li, and S. Lazebnik
2016. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Pp. 5005–5013.

Wen, T.-H., M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young
2015a. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods for Natural Language Processing (EMNLP).*

Wen, T.-H., M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young
2015b. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods for Natural Language Processing.*

Wen, T.-H., D. Vandyke, N. Mrkšić, M. Gasic, L. M. R. Barahona, P.-H. Su, S. Ultes, and S. Young
2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, Pp. 438–449.

Whitney, D., E. Rosen, J. MacGlashan, L. L. Wong, and S. Tellex
2017. Reducing errors in object-fetching interactions through social feedback. In *International Conference on Robotics and Automation, Singapore, May.*

Williams, E. C., M. Rhee, N. Gopalan, and S. Tellex
2017. Learning to parse natural language to grounded reward functions with weak supervision. In *AAAI Fall Symposium on Natural Communication for Human-Robot Collaboration.*

Williams, J. D. and G. Zweig
2016. End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269.*

Williams, R. J.
1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, Pp. 5–32. Springer US.

Wong, Y. and R. J. Mooney
2007. Learning Synchronous Grammars for Semantic Parsing with Lambda Calculus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Pp. 960–967.

Woodward, M. and C. Finn
2017. Active one-shot learning. *Computing Research Repository*, arXiv:1702.06559.

Xiao, F., L. Sigal, and Y. Jae Lee
2017. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 5945–5954.

Yan, C., D. K. Misra, A. Bennett, A. Walsman, Y. Bisk, and Y. Artzi
2018. CHALET: cornell house agent learning environment. *CoRR*, abs/1801.07357.

Yang, B., J.-T. Sun, T. Wang, and Z. Chen
2009. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD'09*. ACM Press.

Yang, S., Q. Gao, C. Liu, C. Xiong, S.-C. Zhu, and J. Y. Chai
2016. Grounded semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Pp. 149–159.

Young, S., M. Gašić, B. Thomson, and J. D. Williams
2013. POMDP-based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5):1160–1179.

Young, S., M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu
2010. The Hidden Information State Model: A Practical Framework for POMDP-based Spoken Dialogue Management. *Computer Speech and Language*, 24(2):150–174.

Yu, Y., A. Eshghi, and O. Lemon
2016. Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, P. 339.

Yu, Y., A. Eshghi, and O. Lemon
2017. Learning how to learn: an adaptive dialogue agent for incrementally learning visually grounded word meanings. In *ACL*.

Zettlemoyer, L. and M. Collins
2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*.

Zhang, S. and P. Stone
2015. CORPP: Commonsense Reasoning and Probabilistic Planning, as Applied to Dialog with a Mobile Robot. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*.