Grounded semantic parsing using captioned videos

by

Candace Cheronda Ross

B.S., Howard University (2015)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Signature redacted

Author ....
Department of Electrical Engineering and Computer Science
May 23, 2018

Signature redacted

Certified by ....
Boris Katz
Principal Research Scientist, MIT CSAIL
Thesis Supervisor

Signature redacted

Accepted by ....
Leslie Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Grounded semantic parsing using captioned videos

by

Candace Cheronda Ross

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2018, in partial fulfillment of the requirements
for the degree of
Masters of Science in Electrical Engineering and Computer Science

Abstract

We develop a semantic parser which is trained in a grounded setting using pairs of videos captioned with sentences. This setting is both data-efficient requiring little annotation and far more similar to the experience of children where they observe their environment and listen to speakers. The semantic parser recovers the meaning of English sentences despite not having access to any annotated sentences and despite the ambiguity inherent in vision where a sentence may refer to any combination of objects, object properties, relations or actions taken by any agent in a video. We introduce a new corpus for grounded language acquisition. Learning to understand language, turn sentences into logical forms, by using captioned video will significantly expand the range of data that parsers can be trained on, lower the effort of training a semantic parser, and ultimately lead to a better understanding of child language acquisition.

Thesis Supervisor: Boris Katz
Title: Principal Research Scientist, MIT CSAIL

Contents

Acknowledgements

This project is joint work with Andrei Barbu, Yevgeni Berzak and Battushig Myanganbayar and my advisor Boris Katz. I want to thank them for their investment in seeing this work through to the finish and for their investment in me. To the people who are not in my lab but still play a critical roles in my life at MIT such as Gloria Anglón, Leslie Kolodziejski and Mandana Sassanfar, I want to thank you for your constant support. You truly are unsung heros. To my parents, sister and brother, thank you for believing in my ability to succeed from childhood. I love you all dearly. This MS thesis marks the conclusion of one chapter in the journey to the PhD.

Chapter 1

Introduction

1.1 Motivation

Children are remarkably rapid language learners. In the first three years of life, children acquire a basic knowledge of their community's language. During this learning process, they are exposed to many nonlinguistics cues such as objects and actions that they see while hearing utterances (Pinker, 1979). Children map linguistic input in the form of natural language to internal meaning representations. While learning language, children do not require direct validation when they misinterpret or incorrectly produce language. Many linguists believe children judge whether utterances are correct based on the response from adults, not from being explicitly corrected (Gold, 1967; Baker, 1979).

The accuracy of meaning representations comes only from the environment (e.g. "Am I getting the response I expect from adults around me? If so, I probably have the correct meaning.") Many current computational models learn differently from children. These models are often explicitly told whether they are correct, which is starkly different from the validation from other speakers and the environment that children use. Because children judge correctness based on understanding, we want to model language learning in this way in machines as well.

If we want a computational model for child language acquisition, the model should be cognitively plausible. There are many criteria that should be considered for cognitive plausibility.

- **modality of data:** children are presented with natural language along with other perceptual inputs such as visual stimuli
- **magnitude of input data:** How many input sentences is the model getting during learning? Is it significantly more than a child receives?
- **data presentation:** children hear sentences incrementally and these sentences are generally not repeated; in addition, children are not given sentences annotated with structure (Clark and Lappin, 2012) and still acquire meaning
- **validation process:** children determine whether they are correctly understanding and generating language based on the environment and interactions with speakers
- **joint learning of syntax and semantics:** children exhibit syntactic bootstrapping, where syntax is used in learning words such as verbs and acquired with semantics (Naigles and Hoff-Ginsberg, 1995)

While we aim to incorporate other criteria in the future, we focus in particular on the modality of data the model receives, the validation process used during learning and joint learning of syntax and semantics for the words in our input data.

There are two popular classes of models that are used for language acquisition. Connectionist models such as neural networks that are broadly inspired by the basis that learning involves activations being propagating in the brain. Another class of models are probabilistic models that use Bayesian or minimum description length (MDL) approaches to learning. These models balance learning the most descriptive grammars that minimize length or complexity. Semantic parsers are the type of model for language acquisition that often exist in both probabilistic and neural approaches. These parsers take natural language as input and produce meaning representations. These meaning representations are a rich encoding of components of the sentence. Meaning representations can be written with many different encoding formats, such as Abstract Meaning Representation (AMR) (Banarescu et al., 2013), Lambda Dependency-Based Compositional Semantics (λ-DCS) (Liang, 2013), and typed-lambda calculus systems (Zettlemoyer and Collins, 2005). We can refer to these meaning representations in the above formats as logical forms. These forms encode the objects, agents, actions and interactions conveyed by the sentence. For example, the logical form

$$\lambda x.\text{person}(x) \wedge \text{in}(x, z) \wedge \text{jeans}(z) \wedge \text{walks}(x) \wedge \text{near}(x, y) \wedge \lambda y.\text{person}(y)$$

refers to two agents, x and y and their interaction, where agent x walks near agent y.

The production of logical forms makes semantic parsers an ideal model for mapping parts of the form to agents, objects and interactions in perceptual input. If we saw Figure 1-1 depicting a scene a corresponding to the logical form above, we could map agent x onto the man on the left in the jeans and leather jacket and agent y onto the man in the trenchcoat. If the input were a video instead of an image, even higher level inferences could be made such as the nature of the interaction and social cues.

Visual context is important in human language processing; Tanenhaus et al. (1995) demonstrates in an experimental study that participants eyes mapped to objects as they were mentioned in sentences. The correlation between language and visual input has historically been unexplored in semantic parsing. Most traditional semantic parsers only use natural language during learning and are fully supervised. This means reasoning about sentence meaning is based solely on labeled forms provided as input alongside sentences and no visual input is provided. For instance, one input example might be the pair (sentence: `Dogs run at the park`, parse: $\lambda x.\text{dog}(x) \wedge \text{run}(x) \wedge \text{at}(x, y) \wedge \lambda y.\text{park}(y)$). While the sentence may be correctly

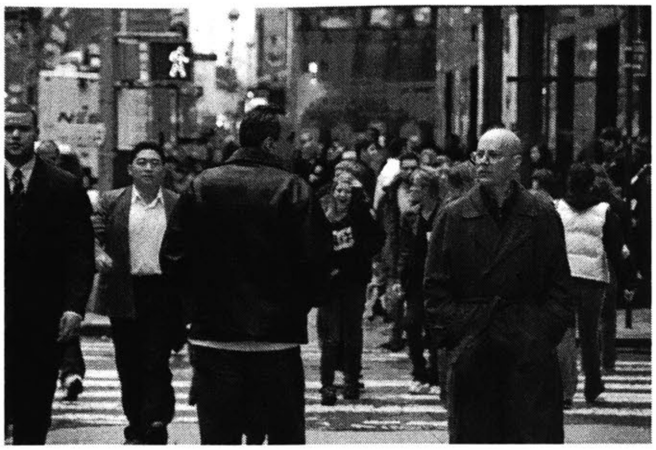


Figure 1-1: Example of two people walking past each other.

parsed, the model never learns that dogs are four legged animals that range in size and might be seen on a leash; this knowledge may have been provided if the sentences had corresponding visual input. In addition to the lack of visual input, the parser has direct feedback during learning because the hypothesized logical forms can be compared to the target logical form from the input. Children do not receive sentences annotated with sentence structure during learning (Clark and Lappin, 2012).

Even semantic parsers that incorporate perception and learn in a weakly supervised manner still differ significantly from the kind of data that children are usually exposed to. Matuszek et al. (2012) uses images of multiple objects and sentences describing subsets of objects in the image as input. The task of the parser is to produce a logical form that can be executed to visually select the correct objects. The computer vision task is straightforward as the objects are rigid and visually salient. In addition, images do not have a temporal component. This constrains the range of linguistic input being learned and makes the task of mapping language to concepts more different from children who learn actions and how agents perform actions.

Another weakly supervised task that of Artzi and Zettlemoyer (2013), where commands in natural language and a robot simulator serve as the input. The accuracy of the parse is determined by comparing the end state of the simulator to the target end state. This task introduces actions but is noiseless. There is not ambiguity in the input in the same way children receive ambiguity through things like object occlusion. Here, I develop a model which does not have these shortcomings, by learning in a dynamic environment with little supervision.

Weak supervision is important both for improving models of child language acquisition and for expanding the breadth of machine learning tasks. One application of a grounded, weakly supervised model are robots deployed in homes. In this example, robots would need to learn the user's language and be able to grounded words to physical objects to be helpful. Ideally, these robots would also interact with humans to expand their knowledge of natural language. Provided explicit meaning representations in this instance would be infeasible. If they approached learning like children, who interacting with the environment to acquiring new conceptual meanings and get feedback on acquisition when they do not get the desired response to an utterance (Gold, 1967), robots would learn more rapidly and robustly.

We want to expand semantic parsers to use the perceptual input similar to children during learning and learn. The goal of this proposal is to present a weakly supervised semantic parser that learns without any labeled logical forms by grounding language in perception.

1.2 Research Problem

We present a model for a ***weakly supervised grounded semantic parser***. The model takes as input short videos along with sentences that are true of those videos and computes the likelihood that the parser's output (hypothesized logical forms) describes the video. The model is significant in that it is the first grounded semantic parser that uses visual input that is a modality similar to children during the learning process.

In our model, the labeled logical forms for the sentences are never provided during training and many of the sentences are linguistically ambiguous. Therefore, reasoning about the visual input is necessary for learning language. There are similarities between our model and child language acquisition, given that children observe the environment, hear what speakers say and make inferences about language. However, this is not a perfect mapping to children, as our model is a passive observer without an ability to directly interact with other speakers. We intend to include active observers in future iterations.

There are many candidate grammar formalisms that can be used for the semantic parsing; we chose to use the Combinatory Categorical Grammar (CCG) (Steedman, 1996, 2000). CCGs are composed of a lexicon, which pairs words/phrases with semantic and syntactic categories, and combinatory rules, which describe how lexical entries can combine to form a complete parse. CCGs jointly learning syntax and semantics, which is one of the criterion mentioned above for cognitively plausible models.

We use the Sentence Tracker framework of Siddharth et al. (2014) and Yu et al. (2015) to infer the accuracy of the hypothesized logical forms from the CCG. The Sentence Tracker maps the components of the logical forms to agents, actions and interactions in the videos and produces a likelihood. The better the mapping between logical form and video, the more likely it is that the model has learned the correct representation. The Sentence Tracker never sees the target logical form and instead uses the environment as a means for determining language understanding. This is another criterion mentioned, where children use the environment around them to infer their understanding of language.

To train and evaluate our parser, we created a grounded semantic parsing dataset for our model consisting of sentences describing videos. We recorded videos and used Amazon Mechanical Turk to gather sentences describing these videos. Our completed dataset contains 1733 videos and 764 sentences. We intend for this dataset to serve as the benchmark for grounded language acquisition and our dataset will therefore be publicly available.

1.2.1 Contributions

To summarize, *our task is to construct a grounded semantic parser that learns in a manner more similar to children by using captioned videos for weak supervision.* The contributions of the work described are as follows. 1) We construct a semantic parser that learns in a manner more similar to children than many existing semantic parsers. One key reason is because validation comes exclusively from a training signal from videos instead of labeled training examples. 2) We demonstrate how to resolve visual and linguistic ambiguities at training time in a manner that can be adapted to other semantic parsers. 3) Our model of child language acquisition is presented data in a similar modality, both linguistic and visual, that children receive. We demonstrate how a small number of annotated sentences can be used to improve performance. 4) We present a large corpus that is the first for grounded semantic parsing using captioned videos. This corpus will be publicly available.

1.3 Thesis Roadmap

Chapter 2 describes existing literature related to semantic parsing and language grounding. This lays the foundation for where our work fits into language acquisition from the perspective of multiple fields. Chapter 3 provides background on semantic parsing using CCGs and the use of sentence trackers to make inferences about sentences and videos. Chapters 4-6 frame describe our dataset, the experimental setup and the results from our different benchmark models and the full grounded model. In Chapter 7, we discussion both challenges of our approach and future directions we intend to pursue.

Chapter 2

Related Work

The following sections describe existing literature related to this thesis. First, we describe the use of grammar induction for modeling language acquisition. Next, we discuss combinatory categorical grammar (CCG) induction for semantic parsing, which is the model we use in our approach. We then discuss grounded language models for visual reasoning, which are important for how we infer about the videos in our input. 2.4 presents psycholinguistic and cognitive science literature that provide a basis for how children learn language and inspire learning approaches in our model.

2.1 Grammar Induction

Grammar induction is the process of learning rules and productions for a language. One of the earliest theories on grammar induction comes from Gold (1967), which posits that learning a grammar is similar to a metric of the ability to "speak the language". Gold presents results on the learnability of languages in a deterministic domain with data presentation of a *text* that presents positive examples only (which are strings in the language) and an *informant* that presents positive and negative examples. Children primarily receive only positive examples, which means natural languages are not learnable in this framework. However, Gold did lay the foundation was laid for grammar induction in a probabilistic setting that could be more similar to children.

Stochastic grammar induction provides an additional layer of inference. In a model where we use a text, it can be difficult to form hypotheses about what strings are not in the language. For instance, assume strings can be composed of symboled from the alphabet $\Sigma = \{a, b\}$ and we have seen the training examples presented *a, aa, aaa, aaaa, aaaaa, aaaaaa, aaaaaaa*} thus far. If the text continued to present strings composed only of the character *a*, it might be plausible to believe the strings of the form a^* in languages in the grammar and all others are not. However, because the identifying languages in the grammar is deterministic, the lack of seeing a *b* does not provide evidence that these strings are not in the grammar.

If we move to a stochastic realm, even without providing explicit negative examples of the form b^*, the absence of evidence can become evidence of absence. As more examples are presented and none contain the character *b*, the likelihood of the grammar accepting only strings composed of *a*'s becomes very high. This allows for additional conclusions to be drawn about the grammar (Angluin, 1988; Clark and Lappin, 2012). Given the power of probabilistic grammars for learning languages, we use probabilistic grammar induction as the basis for our model of language acquisition.

2.2 Semantic Parsing Using CCGs

Semantic parsing is the process of mapping natural language to meaning representations. Semantic parsing is through the induction of Combinatory Categorical Grammars (CCGs) (Steedman, 1996, 2000) is the approach we use in this thesis. CCG-based parsing is an approach that has cognitively plausibility based on many of the criteria discussed in Chapter 1.1. Early approaches to CCG induction were often used for syntactic parsing, where words and phrases in sentences were paired with syntactic categories (Watkinson and Manandhar, 2000; Clark and Curran, 2003). Many of these approaches also required full parse trees as input.

Currently, most CCGs jointly learn syntax and semantics. While a large improvement over prior models in terms of captured linguistic content, these models still required full parse trees (Clark and Curran, 2003) Later models with joint syntax/semantics improve by only requiring labeled logical forms (Zettlemoyer and Collins, 2005, 2007). This reduction in supervision meant less required training data; as parse trees are expensive, this served as a significant improvement.

2.2.1 Grounded Approaches

Training algorithms for inducing CCGs moved toward using weaker supervision, where labeled logical forms were not needed. The weakly supervised models often incorporate context and perceptual input during learning. These training algorithms are the closest to how children learn language, as we know they do not receive explicit correction when producing utterances and use non-linguistic cues during learning (Baker, 1979; Pinker, 1979). One such approach to weak supervision is an execution model. Artzi and Zettlemoyer (2013) use natural language commands and a simulated robotic environment for grounded semantic parsing. Sequences of commands are parsed and executed and the simulation's final state is compared to the goal final state, which serves as validation for the model. Because the model uses a simulated world, there is no perceptual noise introduced. In addition, the robot is constrained by the actions it can take; this constrains the simulated world and by extension the space of possible parses. Execution models are seen in semantic parsing outside of CCGs as well. (Berant et al., 2013) use an execution model for parsing; instead of annotating database queries, the answers are annotated. The model then searches over logical forms that produce the given answer. Our setting produces far more ambiguity as we do not ask a question or produce an explicit response.

There is another approach using weak supervision where the model is trained through an object selection task (Matuszek et al., 2012). The input data are images paired with sentences. In their paper, the task is object selection. Each sentence describes a subset of objects in an image and, if correctly parsed, the logical form should select the correct objects. Their approach recognizes shapes and colors but does not incorporate higher-level visual information like actions and agent interaction. By comparison, our model uses far weaker supervision. The object selection task has an absolute answer during validation (i.e. were the correct set of objects selected?). In our approach, the validation is simply a likelihood of the correct agents, actions and objects being described. This means our system learns from estimation and not absolute correctness. Children also estimate accuracy based on response from speakers (Baker, 1979). Our model is therefore a much closer proxy to children learning language.

There are unsupervised models for semantic parsing as well (Goldwasser et al., 2011). The authors argue that scaling up semantic parsing is constrained by supervision and therefore present a model that does not require labeled logical forms or any form of external validation. While this is a valid claim for the goal of semantic parsing on a diverse set of data, we aim for a model of child learning. Because children have access to at least some form of weak validation, we decided instead on a weakly supervised model instead of an unsupervised model.

2.3 Compositionality

Language is inherently combinatorial and compositional: sounds compose to forms words and words compose to form sentences. There are models that exploit the compositionality of language for different tasks such as video reasoning and caption generation. Siddharth et al. (2014) and Yu et al. (2015) present models of a sentence tracker that takes a video v and conjunction of predicates l and produces the likelihood that the predicates describe the video, $P(l|v)$. The predicates are mapped to agents, actions, and interactions in the video and the model learns the meaning of the predicates, which correspond to natural language, and attributes of the video clip. Applications of this model include include action recognition, video retrieval and sentence generation, and language disambiguation. While the applications are useful in using the compositionality of language and its mapping to visual input, the model does not actually make use of natural language. Sentences are required to be input as conjunctions of predicates, which requires laborious annotations.

Another model is that makes use of this compositionality is Berzak et al. (2015), which disambiguated between multiple meanings of a sentence by making inferences about a corresponding video. The model provides multiple interpretations of ambiguous sentences, presented as logical forms, and disambiguates using corresponding video input. Again, annotations of the sentences are required for disambiguation. Our model also makes use of a sentence tracker but does not need or use any explicit annotations. Inference begins at parsing the natural language sentence and inherently relies on the sentence tracker for validation. This more closely couples the inference between language and vision.

2.4 Psycholinguistics and cognitive science

Psycholinguists and cognitive scientists explore how children learn language and way to model their learning computationally. Children learn language without negative examples in the grammar and do not require being explicitly corrected when they produce ungrammatical utterances (Gold, 1967; Baker, 1979). Children decide on the correctness of utterances based on the response received from adult. This means child learning is very different from many computational models that give explicit feedback about the meaning extracted from natural language. Our model is based on this theory, where the learner uses perceptual input and there is not direct feedback about accuracy.

The integration of linguistic and visual information has also been studied (Tanenhaus et al., 1995): eye movements were recorded while participants received spoken commands about physical objects in front of them. Their eyes mapped to objects as they were being referenced in the commands. This study supports previous research that context, in this case visual, is important for language comprehension. It also provides a basis for our model mapping entities in hypothesized meaning representations to objects in videos.

There is also work on cognitively-plausible semantic bootstrapping, where a Bayesian model is provided with natural language sentences and meaning representations and induces a grammar (Abe, 2017). This approach assumes that children have access to the structural representation of some utterances, which is a semantic or conceptual content. In addition, this model jointly learns syntax and semantics in a manner similar to children. However, this model uses labeled logical forms (with multiple forms per sentence to represent ambiguities). There is no incorporation of perceptual input during learning, which is starkly different than the input presented to children during learning. Our model is similar to the Bayesian model described above, but moves closer to cognitive plausibility in the input.

Chapter 3

Background

In this chapter, we provide background on semantic parsing through induction of a combinatory categorical grammar (CCG). A trained CCG consists of a lexicon and combinatory rules. We also describe sentence tracking, which is a framework for measuring the likelihood of a sentence describing a video. We integrate sentence tracking into the learning process of our parser.

3.1 Combinatory Categorical Grammar (CCG)

Combinatory Categorical Grammar (CCG) is a grammar formalism that maps from sentences to meaning representations (Steedman, 1996, 2000). CCGs learn a lexicon Λ comprised of entries that pair words and phrases with syntactic and semantic catgories. In our implementation, we build on the framework of (Artzi, 2016). Combinatory rules are used to define how the lexical entries can combine to form parses.

3.1.1 Lexicon

The lexicon is comprised of entries that map words and phrases to syntactic and semantic categories. For example, the lexical entry for the word dog is

$$\text{dog} :- N : \lambda x.\text{dog}(x)$$

where the syntactic category is N and the semantic category is $\lambda x.dog(x)$. The syntactic categories used in CCGs often differ from categories in syntactic parsers such as part-of-speech tags. The categories instead describe the syntactic relationship between the entries and the ways they can combine.

Many lexical entries have similar underlying structure. Factored lexicons are used to exploit this similiarity by decomposing entries into a set L of *lexemes* that contain words and tokens and a set T of *templates* that contain syntactic and semantic categories (Kwiatkowski et al., 2011). These lexicons are generally more compact than non-factored lexicons.

To expand the lexicon, one approach is to use a process called *GENLEX*. Depending on the level of supervision, *GENLEX* uses some combination of existing lexemes and templates, words/phrases from the input sentence, and lexical entries from the labeled logical form (depending on the level of supervision) to propose new lexical entries for the parse tree (Zettlemoyer and Collins, 2005, 2007). The process produces a large set of candidate entries, many of which are not relevant for the input sentence. To prune entries, the candidate parses produces from the generation are validated and scored. Only entries from the top-scoring parses are kept and added to the lexicon.

Many different encoding formats are used for lexical entries such as Abstract Meaning Representation (AMR) (Banarescu et al., 2013), Lambda Dependency-Based Compositional Semantics (λ-DCS) (Liang, 2013), and typed-lambda calculus systems (Zettlemoyer and Collins, 2005). In many CCG implementations, the lexicon is represented using typed lambda calculus. Types are used to define the scope of objects. The two most common types are *e* and *t*, which represent entities and truth-values. Many system implement a type hierarchy such that additional types like *p* (person) can inherit types like *e*. There are four main expressions used to build lambda calculus representations:

- logical constants: defined objects, such as $person, dog, table$
- variables: abstraction over objects, e.g., $pick\_up(x, y)$ where x and y are variables
- literal: function applications, where arguments are applied to predicates; from example above, $pick\_up$ is a predicate and the entire phrase $pick\_up(x, y)$ is a literal
- lambda terms: expressions with lambda operators where variables are bound, e.g., $\lambda x.y\ pick\_up(x, y)$

The next section will explore operations to combine these expressions to make complete parses.

3.1.2 Combinatory Rules

CCGs use combinatory rules to describe how lexical entries can combine to form parses. Unlike some other types of grammar induction, CCGs do not learn rules; the rules are provided at training time. We describe some rules that are used in CCGs below.

1. *functional application rule:* This is a basic way to lexical entries to combine. Functional application can go in forward and backward directions. For instance, an example lexical entry (with syntax only)

   the :- *NP/N*

   combines with an entry of type *N* on the right and forms a new entry of type *NP*. This is an example of forward application. There is also backward application, where a syntactic type $A\backslash B$ combines with an entry of type A and forms type *B*.

2. *type-raising rule:* Type raising is used to change the category of a lexical entry. For example, lexical entries with syntactic type ADJ can be changed to *N/N*. This allows for a more compact lexicon.

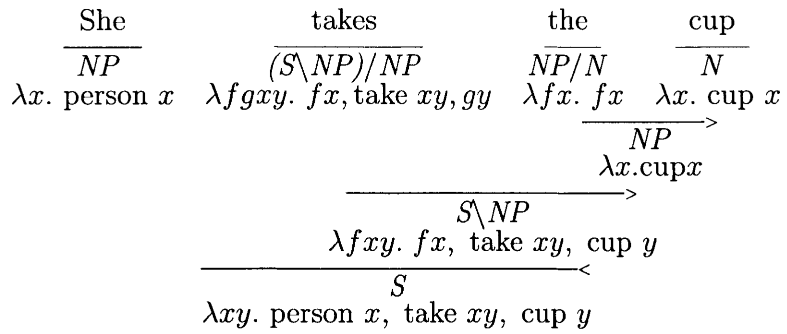


Figure 3-1: A simple sentence parsed into a lambda-calculus expression using a CCG-based grammar. The parse is determined by the lexicon that associates tokens with syntactic and semantic types as well as the order of function applications.

An example of a CCG parse is presented in Figure 3.1.2.

3.1.3 Probabilistic CCGs

CCGs often produce multiple parses for an input sentence. Probabilistic CCGs (PCCGs) are used to rank and select the most likely among the parses. One way to rank parses is to learn a d-dimensional feature vector ϕ and parameters θ where $\theta \in \mathbb{R}^d$. Features typically examine the parse tree and lexical components of the logical form (root of the tree). Examples of features include counting the number of times a lexical entry was used in the parse tree and counting the number of words in the sentences skipped during the parsing. Given a set $\hat{Y}$ of hypothesized parses, the optimal parse $y*$ is:

$$y* = \underset{\hat{y} \in \hat{Y}}{argmax}\ \theta \cdot \phi(x, y)$$

3.1.4 Types of Supervision

In early models of CCGs, the input provided the sentence and parse tree T. In current literature, a fully supervised model usually refers to the sentence and labeled logical form as input; the parse tree is latent. The labeled logical form is used during *GEN-LEX* to propose candidate entries and during parameter estimation for validating the hypothesized parses. There are models that reduce the amount of supervision by removing the labeled logical form. Approaches can either remain completely unsupervised, providing only the input sentence, or use a form of weak supervision. Weakly supervised approaches often are often grounded in a modality that can provide some context for the model to learn and validate its beliefs.

3.2 Sentence Tracker

Sentence Trackers are an approach used to make joint inferences about visual and linguistic input (Siddharth et al., 2014; Yu et al., 2015). The Sentence Tracker takes a logical form l and video v as input and produces the likelihood that the logical form describes the video, $P(l|v)$. We use the Sentence Tracker approach to score parse-video pairs from the CCG-based parser. This approach constructs a parse-specific model by extracting the number of participants in the scene described by a caption as well as the relationships and properties of those participants. It builds a graphical model where each participant is localized by a tracker and each relationship is encoded by temporal models that express the properties of the trackers that those models refer to. The representation chosen for the meaning of sentences is constructed to make building the vision system possible. Each target logical form is a lambda expression with a set of binders, whose domain are objects, and a conjunction of constraints that refer to those binders. In essence, this notes how many objects should be present in a scene and what static and changing properties and relationships those objects should have with respect to one another.

The Sentence Tracker creates one Viterbi-based tracker for each participant and, given a mapping from constraints (which appear as the heads in the logical form) to Hidden Markov models, connects each tracker and each constraint together. Constraints are connected to the trackers which correspond to variables that fill their argument slots in the logical form. Given a video v and a parse p, first a large number of object detections are computed for video by lowering the threshold of an object detector. Trackers weave these detections into high-scoring object tracks and constraints verify if the tracks have the desired properties and relations. Inference proceeds jointly between the vision and parse; the parse focuses vision on events and properties that might otherwise be missed.

Trackers are modeled by a maximum-entropy Markov model with a per-frame score, f, the likelihood that any one object detection is true, as well as a motion-coherence score, g, the likelihood that the track connects two detections. Constraints are modeled by an Hidden Markov Model with a per-frame score, h, which observes one or more tracks (a fixed number depending on the arity of the constraint), and a transition function a which determines the temporal sequence of the constraint if any. Given a parse p with L participants and C constraints along with a video v of length T, the sentence tracker computes the likelihood the parse is true of the video according to

$$\begin{aligned}\max_{J,K}\sum_{l=1}^{L}\left(\sum_{t=1}^{T}f(b^{t}_{j^{t}_{l}})+\sum_{t=1}^{T}g(b^{t-1}_{j^{t-1}_{l}},b^{t}_{j^{t}_{l}})\right)+\\ \sum_{c=1}^{C}\left(\sum_{t=1}^{T}h_c(b^{t-1}_{j^{t-1}_{\gamma^{1}_{c}}},b^{t}_{j^{t}_{\gamma^{2}_{c}}},k^{t}_{c})+\sum_{t=1}^{T}a_c(k^{t-1}_{c},k^{t}_{c})\right)\end{aligned} \tag{3.1}$$

where J is a set of L candidate tracks each ranging over every hypothesis from the object detector, b is a candidate object detection, K is a set of states, one for each constraint, and γ is a linking function. The linking function is an indicator variable that encodes the structure of the logical form filling in the correct trackers as arguments for the corresponding constraints. Here we present the variant for binary constraints but generalizing to unary or ternary constraints, which are required for learning in the domain we present, is trivially done by extending γ and adding arguments to the appropriate constraint observation functions h_c. The domain of this optimization is the combination of all objects at all timesteps that the logical form can refer to as well as every state of each constraint. The Viterbi algorithm carries out this optimization in quadratic time. The result is a likelihood of the parse being true of a video which we use to create the joint model that supervises the parser with vision.

Chapter 4

Methodology

We present a grounded semantic parser that learns under weak supervision using captioned videos. In the sections below, we broadly describe the task we approach in this these. In addition, we detail the learning process, including how grounding in vision allows our model to learn the syntactic and semantic meanings of words and phrases.

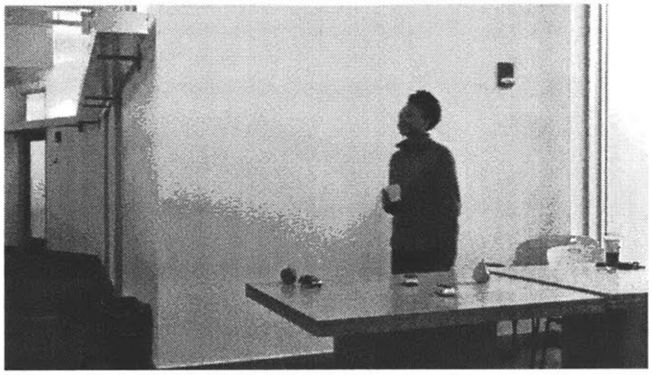
4.1 Task Overview

Given a dataset of captioned videos, D, we train the parameters and lexicon, Λ and θ, of a Combinatory Categorical Grammar (CCG) model for semantic parsing. The lexicon Λ is used to parse the language input and θ is used to score the candidate parses. At training time we perform gradient descent over the parameters θ and employ GENLEX to augment the lexicon Λ. The objective function of the semantic parser is written in terms of a visual-linguistic compatibility between a hypothesized parse, p, and the video, v. This compatibility computes the likelihood of the parse being true of the video, $P(v|p)$. At test time, we take as input a sentence and produce a parse. We compute exact match accuracy by comparing the predicted parse to the ground-truth parse. Test time is the only time the model ever sees the ground-truth parse; no parameters are updated and no entries are added to the lexicon during testing.

For CCG-based (Combinatory Categorical Grammar) semantic parsing, we train in a setting similar to Artzi and Zettlemoyer (2013). We adapt the objective function, training procedure, and feature set to this new scenario. The training procedure involves parsing a sentence and using the visual-linguistic compatibility function to determine if the parse is true of a video. We use the Sentence Tracker developed in Siddharth et al. (2014) and Yu et al. (2015) for the compatability function. Given a parse, the Sentence Tracker produces a targeted detector that determines if the parse is true of a video.

Parses are represented as lambda calculus expressions consisting of a set of binders and a conjunction of sub-expressions referring to those binders. The domain of the variables are potential objects in the videos. In the parse presented in Figure 4-1 example three potential objects are required, represented by the binders x, y, and z. Because of perceptual ambiguities and the huge number of possible referents in any one video we do not explicitly enumerate the space of objects. Instead, we rely on a joint-inference process between the parser and the Sentence Tracker. Intuitively, each sub-expression of the parse asserts a constraint, for example, that one object is approaching another, and the Sentence Tracker verifies these constraints. In Figure 4-1, for example, there is a constraint that whichever objects are bound to x and z, x must be near y, x must be walking, x must be a person, etc.

The objective function assumes that accurate parses should be descriptive of the video. This visual-linguistic inference forms the basis of our model.



Sentence: A woman walks by the table with the yellow cup.
Logical Form: $\lambda x.person(x) \wedge walk(x) \wedge near(x, y) \wedge \lambda y.table(y)$

Figure 4-1: An example sentence and screenshot of a video from the dataset.

4.2 Joint Model

At training time, we jointly learn using both the semantic parser and the language-vision component. At test time, only the parser is used. Two parameters are trained, the lexicon Λ to parse the input sentences and a set of weights θ to score the parses. In both the case of the parser and the associated language-vision component, the lexicon is used to structure inference. The model has three key stages. First, the model attempts to parse the sentence using current lexicon Λ. The Sentence Tracker validates the parses using the approach described in Section 4.1. If valid parses are produced, the model continue straight to the parameter update. If no valid parses are produced, the model uses *GENLEX* to create new lexical entries and attempt to parse the sentence. The created entries from the top scoring valid parses, if any, are added to Λ. At this stage, regardless of whether valid parses were produced, the model updates parameters θ using stochastic gradient descent then goes to the next training example. We detail each step in depth below.

4.2.1 Lexical Generation

We employ a variant of the *GENLEX* procedure from Artzi and Zettlemoyer (2013). *GENLEX* takes as input a validation function — the compatibility between a parse and the video— and the input sentence. This *GENLEX* uses an ontology of predicates, a validation function, and templates from the current lexicon to construct new syntactic and semantic forms. A ground-truth logical form is not required; this is significant because it challenges the parser to learn candidate semantic meanings without ever seeing them in a labeled form.

4.2.2 Parameter Update

For the parameter update, we use stochastic gradient descent; positive and negative parameters updates are computed based on the expectation of getting valid and invalid parses, respectively. Again, this stage either comes directly after parsing with Λ is valid parses are generated at that stage or after *GENLEX*.

4.2.3 Learning Challenges

The joint model must learn these parameters despite several sources of noise. First the vision-language system may simply fail to produce the correct likelihood because computer vision is far from perfect. Often objects that are present in the video are not detected and objects are detected that either are not present in the scene or are present in the scene in a completely different location. Second, an infinite number of possible parses are true of a video because we did not annotate what the sentences refer to in the video. When children learn language, they face this same challenge as they do not have access to bounding boxes or to logical forms. The parse $\lambda x.person(x)$ as well as many other seemingly reasonable parses can be true of a video and cannot be distinguished from the ground-truth parse (which is not available) by the vision component. In addition, these short and true parses like $\lambda x.person(x)$ will have a high likelihood and are likely to have a lot of influence over the parser. This is a far less constrained environment than other approaches to semantic parsing. Yet, by avoiding polysemy and assuming that words have meaning, a parser should still be able to learn in this setting. To this end, we add an additional feature to the semantic parser, the number of constants in the lexical entry of a word, which encourages it to learn to avoid assigning empty semantics, i.e., the identity function, to words. This is motivated by models of communication, such as the Rational Speech Acts model of Frank and Goodman (2012) which states that speakers are unlikely to insert meaningless words.

Third, a practical concern is that computer vision is slow and many evaluations of pairs of parses and videos are required to train a parser. This means training a grounded parser a more expensive task than a directly supervised parser that has access to the labeled form. To overcome, this we construct a provably-correct cache that keeps track of failing subexpressions by taking advantage of a feature of this particular vision-language scoring function: the score decreases monotonically with the number of constraints. This modified semantic parser employing vision-language-based validation learns to parse sentences despite these difficulties.

Chapter 5

Dataset

Existing datasets for semantic parsing are mostly ungrounded, such as Geo880 and Jobs640, or are grounded but do not use videos (such as datasets from Artzi and Zettlemoyer (2013) who use a robot simulator and Matuszek et al. (2012) who use still images of unoccluded objects on a white background). For this reason, we collected and annotated a large dataset comprised of videos and sentences describing the videos. We aimed for videos with multiple agents, objects and actions containing real-world visual phenomena like occlusions. We detail each step of the dataset collection, which includes recording the videos, generating sentences to caption the videos, and annotating the sentences with ground-truth logical forms. We believe this dataset can be useful for researchers interested in semantic parsing under weak supervision and for other language grounding tasks; we will therefore be making it publicly available.

5.1 Video recording

We recorded videos of various agents carrying out one of 15 actions, such as picking up and putting down objects, with one of 20 objects, such as backpacks, fruit, and toys. The objects spanned 10 different colors. In addition, we controlled for 11 spatial relations between objects. Most videos depict multiple agents performing actions leading to additional ambiguity. Videos were filmed in multiple locations with multiple agents but care was taken to ensure that the background and agents are not informative of the events depicted. In total 1733 videos were collected.

5.2 Descriptions of videos and annotations

We used Amazon Mechanical Turk to collect annotations. Participants were given six videos and asked to provide 3 sentences describing each video. We did not specify what to describe to avoid biasing participants. This led to richness of annotations but sometimes lead to annotations that referred to properties of the video that are well beyond the capacities of existing vision system. For example, some sentences referred to the intent of the agent of properties of the camera. We removed such sentences and sentences containing spelling or grammatical errors, although we will release them with the dataset as they may be useful in the future. Examples of excluded sentences are shown in Table 5.1. Two trained annotators created logical forms for each sentence using a set of 75 constraints. A validator checked each parse logical form from the annotators.

5.3 Object detection

Two object detectors were employed using an off-the-shelf person-specific detector, OpenPose, (Cao et al., 2017; Simon et al., 2017; Wei et al., 2016), and an object detector, YoLo (Redmon and Farhadi, 2018), which was fine-tuned on the objects available in this dataset. The object detections are used by the Sentence Tracker to reason to ground predicates in the videos. Many objects in this dataset are small and are handled by humans which leads to a large amount occlusion. This cause pre-trained object detectors failing to recognize most objects available here. We will

Grammatical Errors
One man is walking on the towards to another one.
A man holds a yellow chair at chest level was he walks towards a second man.
A guy in striped shirt cross across the room.
Another man is keep the green color bag on the floor.
Spelling Errors
Two men life the chairs at the same time.
One man is hodling green bag.
Both are wearing switers.
Two men walk up to a man in a plad shirt.
Outside of Vision Scope
She holds up the toy car and looks into the camera.
The man with no book bags is lazy and making his friend hold both.
A man who knows he is being stared at moves his bag to his other hand.
A man works out his right arm.

Table 5.1: Examples of sentences excluded from final dataset. Some excluded sentences contain multiple errors.

provide tuned object detectors for these videos but expect that as computer vision becomes more reliable this will one day not be necessary. Example frames from clips in the dataset along with their correponding captions and logical forms are shown in Figure 6-2. Our dataset for this paper contains 764 sentences.

Chapter 6

Evaluation

6.1 Experimental Setup

We adapted Cornell SFP (Semantic Parsing Framework) developed by Artzi (2016) to jointly reason about sentences and videos. We evaluate our model on the dataset described in Chapter 5. We use 80% (613 examples) of the dataset for training and the remainder (151 examples) for the test set. This split is fixed and used in all experiments below. No sentences or videos occurred in both the training set and test set.

In addition to the input data, CCG-based parsers are seeded with a small number of generic combinations of syntactic and semantic types. We sampled a small number of sentences (less than 1% of total examples) to populate the seed lexicon. The sentences were manually annotated using the 75 predicates described in Chapter 5; the entries composing the parses were added to the seed. These sentences were not included in the training or test sets. There are 100 unique lexical entries in the seed; other similar grounded approaches have a similar number of seed lexical entries (e.g., Artzi (2016) provide 141 possible types). We use the following types in our system:

- *e:* entity; this can refer to any constant in our system

- *p:* people
- *o:* inanimate objects; this is essentially all objects that are not people
- *a:* actions; these are actions that are only carried out by people, such as *pick up* and *drop*
- *t:* truth-value

These types and the 75 predicates described in Chapter 5 are used to form the entries.

In the various experiments below, each hypothesized parse for each sentence is marked as either correct or incorrect, using either direct supervision with the target parse or with a compatibility function using the video depending on the experiment. To generate hypotheses we used a CKY-parser with a beam of 180. The rules used in parsing included type-raising and word-skipping with a fixed cost of 1. We allowed for sloppy parses meaning the parser can skip up to 1 word in each sentence if the initial attempt to parse fails.

The model uses *GENLEX* to create new lexical entries whenever it either fails to parse a sentence entirely or only produces incorrect parses. *GENLEX* has a beam of 180 in generating new entries; only the entry from the top scoring parse during lexical induction is added to the lexicon Λ. If there are not any valid parses during *GENLEX*, no new entries are added and the parser continues to the next training example. During coarse lexical generation in runs that do not provide labeled examples, *GENLEX* uses an ontology for constants to insert in generated templates. This ontology contains predicates that occur in the dataset without semantic or syntactic categories; only the constant and type are provided, e.g. $move :< e, < e, t >>$. We ran each experiment below for 5 epochs and kept the above parameters constant in each run. We use two metrics when reporting results. *Exact matches* are where the predicted parses must match the target parses and *near misses* are where a single predicate in the semantic parse is allowed to differ from the target.

6.2 Results

Figure 6-1 summarizes the experiments and ablation studies performed. First, to establish chance-level performance we trained the model on video-sentence pairs where we shuffled the video labels. Random ground-truth logical forms were assigned to random sentences. This is more powerful than a simple chance-level performance calculation as the parser can still take advantage of any dataset biases. However, performance is very low with F1 scores of 0 and 0.10 for the exact and near miss metrics. Both metrics pose a challenging learning problem and demonstrate the importance of visual input during learning.

For the next baseline, we directly supervised the parser with the target logical forms. When doing so, the model achieved high performance with F1 scores of 0.84 and 0.93 for the exact match and near miss cases. Figure 6-1(b&c) show performance of direct supervision as a function of training set size. Even with the smallest amount of data we trained with (10%, around 60 examples) the model still performed well on the test set that was over twice its size. These results demonstrate the sheer amount of information available to the parser under direct supervision.

To further explore performance of direct supervision, we added noise to the model during training. Doing so simulates the unreliable nature of vision and, to an extent, the ambiguities inherent in vision. Noise was introduced by modifying the validation function that determines if a parse is correct. A certain percentage of the time, the function returned true or false randomly when given a hypothesized logical form. With around 60% noise, performance was 0.22 and 0.39 F1 for the exact match and near miss cases. Figure 6-1(b&c) show performance of the noisy baseline as a function of how much noise was introduced. The decrease in performance under noise is significant and shows the difficulties of learning with uncertainty.

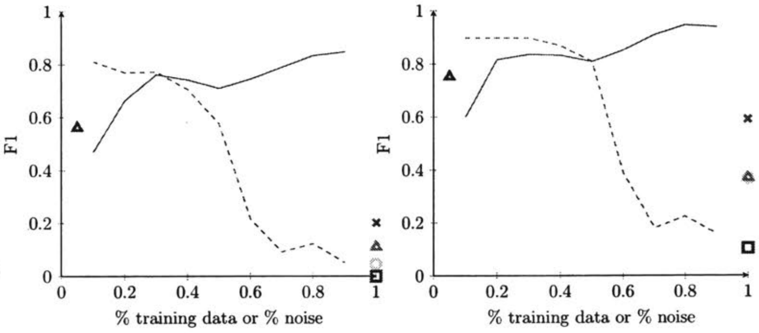
Finally, we trained the grounded parser. The model produced 0.20 and 0.60 F1 scores for the exact and near miss metrics during test. This is far chance performance and corresponds to direct supervision with around 55% noise. There are a number of reasons for why performance is not perfect. First, the evaluation metrics cannot consider equivalences in meaning, just form. A parse may carry the same meaning as the target logical form yet it will be marked incorrect. This is much less of a problem with direct supervision where the preferences that human annotators have for a particular way of expressing the meaning of a sentence can be learned directly. In the grounded case, this cannot be learned and equivalent parses are all equally likely. Second, computer vision is unreliable, e.g., object detectors often fail. Third, vision in the real world is very ambiguous. Predicates like, *hold*, are true in almost every interaction. This makes learning the meanings of words much more difficult resulting in the grounded parser often adding useless entries into the predicted logical forms or substituted one predicate for a similar one. The near miss metric shows that overall the parser learned reasonable logical forms. Figure 6-2 shows six examples from our dataset along with expected and predicted parses, both correct and incorrect.

To understand how much of the performance of the grounded parser comes from a trivial visual correlation like the presence or absence of particular objects versus more complex and cognitively relevant spatio-temporal relations like actions, we ablated the parser. We removed all features other than objects. The resulting grounded parser accepts any hypothesized parse as long as the objects mentioned in that parse are present in the video. This led to a significant performance drop, well below chance level performance on the exact metric, F1 0.05, and nearly half the F1 score on the near miss metric, 0.37. Having a sophisticated vision system, not merely a lookup for the presence and absence of objects, is crucial for learning.

Finally, we sought to explore the performance of the parser when providing a small amount of labeled bootstrapped data. This falls within the realm of semantic bootstrapping, which posits that children has access to structured representations of a portion of their input data. We provided our model with varying amounts of labeled training examples (between 1-10% for different runs) with the remainder of the data being unlabeled and paired with videos as done in the prior example. Within an individual run, we did not varying the labeled examples being provided. The randomly sampled labeled examples were provided to the parser at the end of each epoch.

Precision	Recall	F1
Direct supervision		
0.851	0.84	0.846
0.946	*0.933*	*0.939*
Noisy supervision (60%)		
0.235	0.201	0.217
0.423	*0.362*	*0.390*
Shuffled labels		
0.121	0.102	0.111
0.407	*0.34*	*0.37*
Shuffled videos		
0.000	0.000	0.000
0.106	*0.103*	*0.104*
Object-only vision		
0.051	0.042	0.046
0.387	*0.349*	*0.367*
Vision-language		
0.223	0.183	0.201
0.663	*0.553*	*0.591*
Bootstrapped with supervision (5%)		
0.568	0.556	0.562
0.758	*0.747*	*0.752*

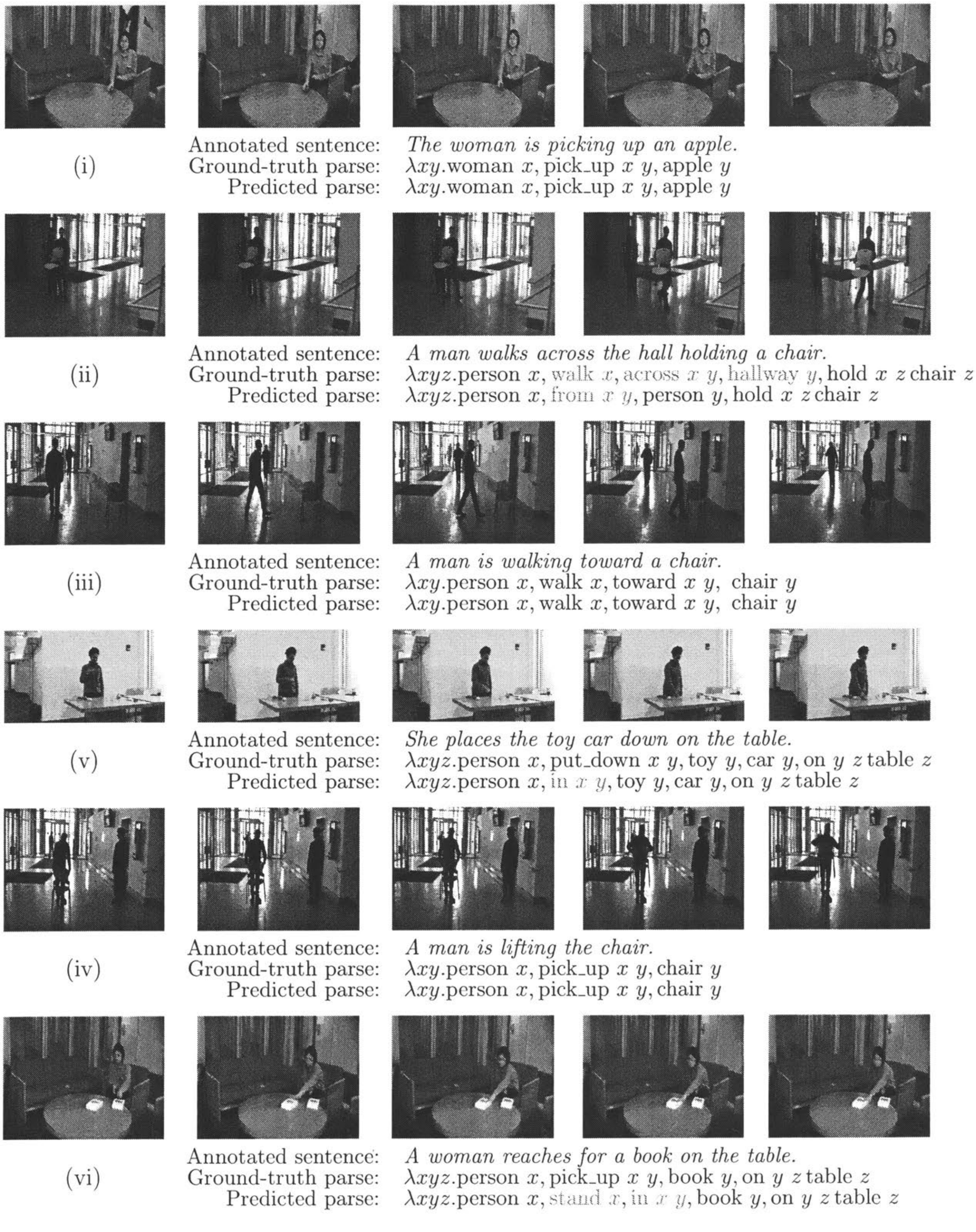
(a) Results from multiple experiments with varied supervision



(b) Exact match

(c) Near miss

Figure 6-1: (a) On the top, exact match results and on the bottom, in *italics*, results for the near miss metric. For *direct supervision*, we provide the actual target logical forms. For *noisy supervision*, when training, a percentage of the time the parser ignores the target and randomly accepts or rejects a parse. For *shuffled videos*, the videos are randomly assigned to sentences. For *object-only*, the vision system consists solely of an object detector and does not consider actions or spatial relations. The full *vision-language* approach learns to parse a significant fraction of the sentences, far outperforming the object-only approach, and usually being within one predicate of the correct answer. (b & c) In blue, *direct supervision* as a function of the amount of training data. In dashed blue, *noisy supervision* uses the whole training set but accepts and rejects parses at random for a given fraction of the time. The red cross is the full vision system while the green o is the lone object detector ablation. The orange triangle *shuffled labels* show chance performance. The blue square shows shuffle videos where the parser receives feedback from the Sentence Tracker on a randomly selected video. The purple triangle shows *bootstrapped weak supervision* where the model is provided a small amount (5% on the graph) of labeled training data. While direct supervision outperforms vision-only supervision the grounded parser closes the gap and operates like noisy direct supervision with roughly 55% noise.



(i)
Annotated sentence: *The woman is picking up an apple.*
Ground-truth parse: $\lambda xy.\text{woman } x, \text{pick\_up } x\ y, \text{apple } y$
Predicted parse: $\lambda xy.\text{woman } x, \text{pick\_up } x\ y, \text{apple } y$

(ii)
Annotated sentence: *A man walks across the hall holding a chair.*
Ground-truth parse: $\lambda xyz.\text{person } x, \text{walk } x, \text{across } x\ y, \text{hallway } y, \text{hold } x\ z\, \text{chair } z$
Predicted parse: $\lambda xyz.\text{person } x, \text{from } x\ y, \text{person } y, \text{hold } x\ z\, \text{chair } z$

(iii)
Annotated sentence: *A man is walking toward a chair.*
Ground-truth parse: $\lambda xy.\text{person } x, \text{walk } x, \text{toward } x\ y,\ \text{chair } y$
Predicted parse: $\lambda xy.\text{person } x, \text{walk } x, \text{toward } x\ y,\ \text{chair } y$

(v)
Annotated sentence: *She places the toy car down on the table.*
Ground-truth parse: $\lambda xyz.\text{person } x, \text{put\_down } x\ y, \text{toy } y, \text{car } y, \text{on } y\ z\, \text{table } z$
Predicted parse: $\lambda xyz.\text{person } x, \text{in } x\ y, \text{toy } y, \text{car } y, \text{on } y\ z\, \text{table } z$

(iv)
Annotated sentence: *A man is lifting the chair.*
Ground-truth parse: $\lambda xy.\text{person } x, \text{pick\_up } x\ y, \text{chair } y$
Predicted parse: $\lambda xy.\text{person } x, \text{pick\_up } x\ y, \text{chair } y$

(vi)
Annotated sentence: *A woman reaches for a book on the table.*
Ground-truth parse: $\lambda xyz.\text{person } x, \text{pick\_up } x\ y, \text{book } y, \text{on } y\ z\, \text{table } z$
Predicted parse: $\lambda xyz.\text{person } x, \text{stand } x, \text{in } x\ y, \text{book } y, \text{on } y\ z\, \text{table } z$

Figure 6-2: Six examples of videos along with target and predicted logical forms showing both successes and failures. Failures are highlighted in red. Note how even errors are similar to the original semantic forms showing that the intended meaning is usually preserved even in these cases.

Chapter 7

Discussion

7.1 Summary

We created a semantic parser that learns the structure of language using weak supervision from vision and then parses sentences without the need for visual input. During training, he model receives videos and sentences describing the videos and during test, the model receives only sentences and produces logical forms. This grounded approach incorporates perception directly into the learning process. Our results show that, while this task is quite difficult, it is possible to learn the semantics of natural language with only visual input as validation. For 20% of the test set, the model produced the exact parse; and for nearly 60% of sentences, the model recovered almost the entire parse without only one incorrect predicate. In many of these cases, the incorrect predicate had a similar meaning to the target predicate and would be difficult to differentiate in a visual setting. Learning by passive observation in this way extended the capabilities of semantic parsers and points the way to a more cognitively plausible model of language acquisition.

7.2 Challenges

Our model still poses several limitations. Evaluating the accuracy of parses depending on a match to a logical form for a sentence parsed by a human is an overly strict criterion. This is a problem that also plagues other approaches such as fully-supervised syntactic parsing (Berzak et al., 2016). Two logical forms may express the same meaning but be written in different ways. For example, if the sentence is `The person walked toward the table`, two different yet equally valid logical forms are

$$(1)\ \text{person}(x) \wedge \text{walk}(x) \wedge \text{toward}(x, y) \wedge \text{table}(y)$$

$$(2)\ \text{person}(x) \wedge \text{approach}(x, y) \wedge \text{table}(y)$$

However, these logical forms differ significantly (highlighted in red). The only overlapping predicates are the objects and, even using our near miss criterion, these sentences still fail because they differ by more than a single predicate. It is not yet clear what an effective evaluation metric is for these grounded scenarios to ensure that meaning is being measured and not merely a perfect match a logical form that could be written multiple ways. Learning in such a passive scenario is hard as correlations between events, every *pick up* involves a *touch*, are very difficult to disentangle.

Another challenge of our model is the static nature in which it learn by observation instead of interaction. Children have the benefit of interacting with speakers as an additional form of validation; if they receive responses that differ from their expectation, they can update their beliefs. In our model, an overgeneralization early on is not easily corrected and propagated through training. For instance, many of these verbs that were mislearned for other verbs with similar meaning could be corrected given the ability for additional feedback.

7.3 Future Work

In the future, we intend to add a mental simulator allowing the learner to imagine scenarios where a predicate might not hold in order to determine if two predicates are correlated in this way and, using Gricean principles, prefer a more specific interpretation for a given video. The sentences selected here were all chosen such that they are true of the video being shown yet much of what people discuss is ungrounded, or at least not grounded in the current visual scene.

We also intend to extend this model to have both a weakly visually supervised mode and an alternate unsupervised parser cost function while automatically determining if a sentence should be grounded visually or not during learning. This will allow our parser to handle both groundable sentences like those in this dataset and more abstract sentences that refer to agents' intentions or future actions. Models of grounded language acquisition, the task that all humans engage in, are far away from operating in a setting that is comparable to that which children experience but this work is a step along that direction. We expect that this work will find applications in robotics where learning to adapt to the specific language of a user while engaging with them is of utmost importance when deploying robots in user's homes.

In addition, we aim to use this model in a task for a robot for simple command following. Showing commands describing simple actions and salient objects, we hope to see if the robot can use this model to both parse to executable commands and dynamically learn the meaning of the actions and objects it sees.

References

2017. Bootstrapping language acquisition. *Cognition* 164:116 – 143.

Dana Angluin. 1988. *Identifying Language from Stochastic Examples*. Ph.D. thesis, Yale University.

Yoav Artzi. 2016. Cornell spf: Cornell semantic parsing framework.

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

C. L. Baker. 1979. Syntactic Theory and the Projection Problem. *Linguistic* 10(4):533–581.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *LAW@ACL*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. *Empirical Methods for Natural Language Processing* .

Yevgeni Berzak, Andrei Barbu, Daniel Harari, and Boris Katz. 2015. Do You See What I Mean ? Visual Resolution of Linguistic Ambiguities. *The 2015 Conference on Empirical Methods on Natural Language Processing* (September):1477–1487.

Yevgeni Berzak, Yan Huang, Andrei Barbu, Anna Korhonen, and Boris Katz. 2016. Anchoring and agreement in syntactic annotations. *arXiv preprint arXiv:1605.04481* .

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Alexander Clark and Shalom Lappin. 2012. Computational Learning Theory and Language Acquisition. *Handbook of the Philosophy of Science* pages 445–475.

Stephen Clark and James R. Curran. 2003. Log-linear models for wide-coverage ccg parsing .

Michael C Frank and Noah D Goodman. 2012. Predicting pragmatic reasoning in language games. *Science* 336(6084):998–998.

E. Mark Gold. 1967. Language identification in the limit. *Information and Control* 10(5):447–474.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, pages 1486–1495.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical Generalization in CCG Grammar Induction for Semantic Parsing. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* pages 1512–1523.

Percy Liang. 2013. Lambda dependency-based compositional semantics. *CoRR* abs/1309.4408.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. ACM, pages 1671–1678.

Letitia R. Naigles and Erika Hoff-Ginsberg. 1995. Input to verb learning: Evidence for the plausibility of syntactic bootstrapping. *Developmental Psychology* .

S Pinker. 1979. Formal Models of Language Learning. *Cognition* 7:217–283.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv* .

Narayanaswamy Siddharth, Andrei Barbu, and Jeffrey Mark Siskind. 2014. Seeing what you're told: Sentence-guided activity recognition in video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*.

Mark Steedman. 1996. *Surface Structure and Interpretation*, volume 1. The MIT Press.

Mark Steedman. 2000. *The Syntactic Process*, volume 1. The MIT Press.

MK Tanenhaus, MJ Spivey-Knowlton, KM Eberhard, and JC Sedivy. 1995. Integration of Visual and Linguistic Information in Spoken Language Comprehension. *Science* 268(5217):1632–1634.

Stephen Watkinson and Suresh Manandhar. 2000. *Unsupervised Lexical Learning with Categorial Grammars Using the LLL Corpus*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 218–236.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *CVPR*.

Haonan Yu, N. Siddharth, Andrei Barbu, and Jeffrey Mark Siskind. 2015. A compositional framework for grounding language inference, generation, and acquisition in video. *Journal of Artificial Intelligence Research* .

Luke Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. AUAI Press, Arlington, Virginia, pages 658–666.

Luke S Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*. pages 678–687.