Grounding language acquisition by training semantic parsers using captioned videos

Candace Ross
CSAIL, MIT
ccross@mit.edu

Andrei Barbu
CSAIL, MIT
abarbu@mit.edu

Yevgeni Berzak
BCS, MIT
berzak@mit.edu

Battushig Myanganbayar
CSAIL, MIT
btushig@mit.edu

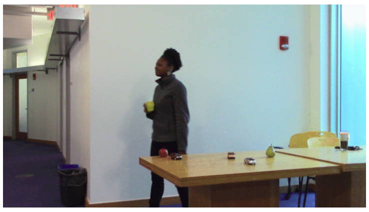
Boris Katz
CSAIL, MIT
boris@mit.edu

Abstract

We develop a semantic parser that is trained in a grounded setting using pairs of videos captioned with sentences. This setting is both data-efficient, requiring little annotation, and 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. It does so 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. For this task, we collected a new dataset for grounded language acquisition. Learning a grounded semantic parser — turning sentences into logical forms using captioned videos — can 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.

1 Introduction

Children learn language from observations that are very different in nature from what parsers are trained on today. Most of the time, rather than receiving direct feedback such as annotated sentences or answers to direct questions, children observe and occasionally interact with their environment. They must use these observations to learn the structure of the speaker's language despite never seeing that structure overtly. This weak and indirect supervision where most of the information is obtained through passive observation poses a difficult disambiguation problem for learners: how do you know what the speaker is referring to in the environment, i.e., what does the speaker mean? Speakers can refer to actions, objects, the properties of actions and objects, relations between those actions and objects, as well as other features in the environment and generally do so by combining multiple features into complex sentences. Moreover, speakers need not refer to the most visually salient parts of a visual scene. Here, we induce a semantic parser by simultaneously resolving visual ambiguities and grounding the semantics of language using a corpus of sentences paired with videos without other annotations.



The woman walks by the table with a yellow cup.
$\lambda xyz.$woman x, walk x, near x y, table y,
hold x z, yellow z, cup z

Figure 1: We develop a semantic parser trained on video-sentence pairs, *without parses*. At inference time a sentence, *without a video*, is presented and a logical form is produced.

The goal of semantic parsing is to convert a natural-language sentence into a representation that encodes its meaning. The parser takes sentences as input and produces these representations – a lambda-calculus expression in our case – that can be used for a variety of tasks such as querying databases, understanding references in images and videos, and answering questions. To train the parser presented here we collected a video dataset, balanced such that the raw statistics of the co-occurrences of objects and events are not informative, and asked annotators on Mechanical Turk to produce sentences that are true of those videos. The parser is presented with pairs of short clips and sentences. It hypothesizes potential meanings for those sentences as lambda-calculus expressions. Each hypothesized expression serves as input for a modular vision system that constructs a specific detector for that lambda-calculus expression and determines the likelihood of the parse being true of the video. The likelihood of the parse with respect to the video is used as supervision for the parser. To

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2647–2656
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

test the parser, we annotated each sentence with its ground-truth semantic parse, but this information is not available at training time.

This process introduces ambiguity. For example, Figure 1 shows a frame from a video annotated with the sentence *"The woman walks by the table with a yellow cup."*, yet the parse, $\lambda x.\ \text{object}(x)$, corresponding to a sentence like *"There exists an object."*, is also true of that video. For a single video there exists an infinite number of true parses that have high likelihood with respect to the vision system because they are indeed indicative of something that is occurring in the video. We demonstrate how to construct a semantic parser that resolves this ambiguity and acquires language from captioned videos by learning to tune the amount of polysemy in the induced lexicon.

This work makes several contributions: We show how to construct a semantic parser that learns language in a setting closer to that of children. We demonstrate how to jointly resolve linguistic and visual ambiguities at training time in a way that can be adapted to other semantic parsing approaches. We demonstrate how such an approach can be used to augment data where a small number of directly annotated sentences can be combined with a large number of videos paired with sentences in order to improve performance. We release a dataset systematically constructed and annotated on Mechanical Turk for joint visual and linguistic learning tasks.

2 Prior work

Learning to understand language in a multimodal environment is a well-developed task. For example, visual question answering (VQA) datasets have led to a number of systems capable of answering complex questions about scenes (Antol et al., 2015). The goal of our work is not to produce answers for any one set of questions, although it is possible to do so from our results; it is instead to learn to predict the structure of the sentences and their meaning. This is a more general and difficult problem, in particular because at test time we do not receive any visual input, only the sentence. The resulting approach is reusable, generic and more similar to the kind of general-purpose linguistic knowledge that humans have. For example, one could use it to guide robotic actions. Al-Omari et al. (2017) acquire a grammar for a fragment of English and Arabic from videos paired with sentences. They learn a small number of grammar rules for a language restricted to robotic commands. Learning occurs mostly in simulation and with little visual ambiguity, and the resulting model is not a parser but a means of associating n-grams with visual concepts.

Siddharth et al. (2014) and Yu et al. (2015) acquire the meaning of a lexicon from videos paired with sentences but assume a fully-trained parser. Matuszek et al. (2012) similarly present a model to learn the meanings and referents of words restricted to attributes and static scenes. Hermann et al. (2017) extend these notions to train agents that learn to carry out instructions in simulated environments without the need for a parser, but do so using simple adjective-noun-relation utterances. Kollar et al. (2013) learn to parse similar utterances in an interactive setting. Wang et al. (2016) create a language game to learn a parser but do not incorporate visual ambiguity or fallible perception.

Berant et al. (2013) describe semantic parsing with execution by annotating answers to database queries. This learning mechanism provides the same results as the one described here: a parser produces the meanings of sentences at inference time without requiring the database, or in our case a video. Databases have far less ambiguity than videos; there is not a temporal aspect to their contents and there is not a notion of unreliable perception. Berant and Liang (2014) learn to parse sentences from paraphrases; one might consider the work here as concerned with visual and not just linguistic paraphrases. Artzi and Zettlemoyer (2013) consider a setting where a validation function involves the dynamic actions of a simulated robot while sentences describe its actions.

3 Task

Given a dataset of captioned videos, D, we train the parameters and lexicon, θ and Λ, of a semantic parser. At training time, we perform gradient descent over the parameters θ and employ *GENLEX* (Zettlemoyer and Collins, 2005) 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 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 without an associated video and produce a semantic parse. We could in principle also take as input the video and produce a targeted parse for that visual scenario. This is a problem similar to that considered by Berzak et al. (2015), but we do not do so here.

We create a CCG-based (Combinatory Categorical Grammar; Steedman (1996)) semantic parser capable of being trained in this setting. To do so, we adapt the objective function, training procedure, and feature set to this new scenario. The visual-linguistic compatibility function is similar to the Sentence Tracker developed in Siddharth et al. (2014) and Yu et al. (2015). Given a parse, the Sentence Tracker produces a targeted detector that determines if the parse is true of a video, which provides a weak supervision signal for the parser.

Parses are represented as lambda-calculus expressions consisting of a set of binders and a conjunction of literal expressions referring to those binders. The domain of the variables are the potential object locations, or object tracks, in the videos. For example, in the parse presented in Figure 1, three potential object track slots are available, represented by the binders x, y, and z. Because of perceptual ambiguities and the large number of possible referents in any one video, we do not explicitly enumerate the space of object tracks. Instead, we rely on a joint-inference process between the parser and the Sentence Tracker. Intuitively, each literal expression of the parse asserts a constraint; for example, if an expression conveys that one object is approaching another, the Sentence Tracker will search the space of object tracks and attempt to satisfy these constraints. In Figure 1, for instance, there is a constraint that for whichever objects are bound to x and z, x must be near y, x must be walking, x must be a person, etc.

4 Model

We develop an approach that combines a semantic parser with a vision system at training time, but does not require the vision system at test time.

4.1 Semantic Parsing

We adopt a semantic parsing framework similar to that of Artzi and Zettlemoyer (2013), although the general approach of using vision as weak supervision for semantic parsing generalizes to other parsers. CCG-based parsing employs a small number of fixed unary and binary derivation rules (Steedman, 2000) while learning a lexicon. In CCG-based parsing, a parser takes as input a sequence of tokens and a lexicon that maps tokens to potential syntactic types and derives parse trees by creating and ranking multiple hypotheses that combine those types together. The syntactic types are richer than other approaches and include forward and backward function application (the forward and backward slash) in addition to the standard syntactic categories. Each derivation has a current syntactic type that is the result of the application of a sequence of rules. To create a derivation, at each step the parser applies each rule to either an individual subderivation or to a pair of subderivations. This process produces multiple hypotheses. Parsing rules are generic, polymorphic, and language-neutral and include concepts like function application and type raising (Carpenter, 1997). The parser accepts a derivation when the tree reaches a single node. We refer to the single node of the parse tree as the logical form. Figure 2 shows a parse starting with tokens and their syntactic types along with each rule being applied.

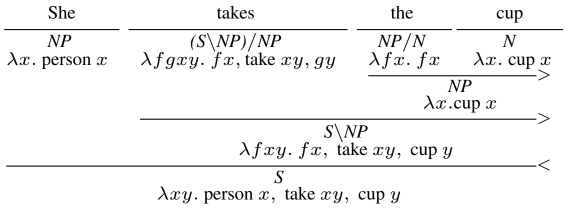


Figure 2: A simple sentence parsed into a lambda-calculus expression using a CCG-based grammar. The parse is determined by the lexicon that associates tokens with syntactic and semantic types as well as the order of function applications. Here, we acquire this lexicon and a means to score derivations.

Semantic parsing with CCGs extends this framework to simultaneously derive a logical form while performing syntactic parsing. Each syntactic rule includes a simple semantic component that manipulates the logical form of its arguments. For example, the forward application rule reduces the syntactic type by applying the syntactic type of the right argument to that of the left, while at the same time performing a lambda-calculus reduction of the semantic types of those same arguments. Concretely, consider a case from Figure 2 where a determiner is attached to a noun, *the cup*. The tokens the and cup are hypothesized to have syntactic types NP/N and N (a function returning NP given an argument on the right side and a noun) and semantic type $\lambda fx.fx$ and $\lambda x.\text{cup}(x)$ (the identity function and a function that adds a cup constraint). These two derivations can be reduced by forward application, denoted by $>$. Both the syntactic and semantic types are applied and reduced, which means the semantics helps guide the syntax. Derivations that produce illegal operations, such as applying an argument to a constant, are forbidden.

Following Zettlemoyer and Collins (2005) and Curran et al. (2007), we adopt a weighted linear semantic parser. For each sentence paired with its hypothesized derivation, this approach computes a feature vector ϕ and a parameter vector θ. Given a sentence s, a parse p, a lexicon Λ, the set of all possible parses for that sentence with that lexicon, $P(s, \Lambda)$, and an n-dimensional feature vector computed for that sentence and parse, $\phi(s, p)$, the parser optimizes

$$\operatorname*{argmax}_{p \in P} \ \theta \cdot \phi(s, p) \qquad (1)$$

to find the best parse p^*. Using a fixed-width beam search, the parser enumerates derivations by choosing a potential syntactic and semantic type for each token from the lexicon and choosing a set of derivation rules to apply. For the i-th training sample d_i, consisting of a sentence d_i^s and a video d_i^v in dataset D and the feature function, the parser finds margin-violating positive, E^+, and negative, E^-, parses, and then uses

$$\theta + \frac{1}{|E_i^+|} \sum_{e \in E_i^+} \phi_i(e, d_i^v) - \frac{1}{|E_i^-|} \sum_{e \in E_i^-} \phi_i(e, d_i^v) \qquad (2)$$

to update the parameter θ. After each sweep through the dataset, the lexicon Λ is augmented using the modified *GENLEX* from Artzi and Zettlemoyer (2013), which does not require the ground-truth logical form. At no point is the logical form needed for updating the lexicon or parameters; we rely instead on a visual validation function to compute the margin-violating examples.

Rather than attempting to learn a fixed lexicon that directly maps tokens to semantic and syntactic parses, we use a factored lexicon like that of Kwiatkowski et al. (2011). This represents tokens and any associated constants separately from potential syntactic and semantic types. For example, the token *chair* is associated with a single constant `chair`; *chair* $\vdash$ [`chair`]. In addition to the token-constants pairs, there exists a list of pairs of syntactic and semantic types along with placeholders for constants; in the case for *chair*, a useful type might be $\lambda v.[N : \lambda x.\texttt{placeholder}(x)]$. When parsing, each token is applied to a potential syntactic and semantic type and the derivation proceeds from there. The factored lexical entries allow for far greater reuse; the model learns a small number of constants that a word can imply separately from a small number of syntactic and semantic types for any word. The weighted linear CCG-based parser searches over potential lexical entries, applying the token to different syntactic and semantic types and over multiple hypotheses for which rule should be applied. At training time, in order to learn a reasonable lexicon and set of parameters, a supervision signal is required to validate candidates. We provide that supervision using the vision system described below.

4.2 Sentence Tracking

To score a video-parse pair, we employ a framework similar to that of Yu et al. (2015). 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 an object tracker and each relationship is encoded by temporal models that express the properties of the trackers that those models refer to. The parser's output representation is chosen 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 which 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 to Hidden Markov models (HMMs), connects each tracker and each constraint together. Given a video v and a parse p, first a large number of object detections are computed for the video by using a low confidence threshold of an object detector. Trackers weave these bounding-box detections into high-scoring object tracks and use constraints to verify if the tracks have the desired properties and relations. Inference proceeds jointly between vision and the parse to allow the parse to focus the vision component on events and properties that might otherwise be missed.

Understanding the relationship between a sentence and a video requires finding the objects that the sentence refers to and determining if those objects follow the behavior implied by the sentence. We carry out a joint optimization that finds objects whose behavior follows certain rules. For clarity, the two steps are presented separately, while we find the global optimum for a linear combination of Equation (3) and Equation (4). Object trackers are 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 bounding boxes selected between frames refer to the same object instance. Given a parse p with L participants and a video v of length T, Equation (3) shows the optimization where J is a set of L candidate tracks ranging over every hypothesis from the object detector and b is a candidate object detection.

$$\max_{J} \sum_{l=1}^{L} \left(\sum_{t=1}^{T} f(b^{t}_{j^{t}_{l}}) + \sum_{t=1}^{T} g(b^{t-1}_{j^{t-1}_{l}}, b^{t}_{j^{t}_{l}}) \right) \tag{3}$$

Determining if an object track follows a set of behaviors implied by a sentence is done using a collection of HMMs. Each has a per-frame score h that observes one or more objects tracks, depending on the number of participants in the behavior being modeled, and a transition function a that determines the temporal sequence of the behavior. Given a parse p with C behaviors, also termed constraints, along with a video v of length T, Equation (4) shows the optimization where K is a set of states, one for each constraint, and γ is a linking function.

$$\max_{J,K} \sum_{c=1}^{C} \left(\sum_{t=1}^{T} h_c(b^{t-1}_{j^{t-1}_{\gamma^1_c}}, b^{t}_{j^{t}_{\gamma^2_c}}, k^{t}_{c}) + \sum_{t=1}^{T} a_c(k^{t-1}_{c}, k^{t}_{c}) \right) \tag{4}$$

The linking function is an indicator variable that encodes the structure of the logical form thereby filling in the correct trackers as arguments for the corresponding constraints. The exposition above presents a variant using binary constraints that is trivially generalized to n-ary constraints by extending γ and adding arguments to the appropriate constraint observation functions h_c. The domain of the optimization problem 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 time linear in the length of the video and quadratic in the number of detections per frame. The result is a likelihood of the parse being true of a video. This is used to create the joint model that supervises the parser with vision. The tracker can also produce a time series of bounding boxes that make explicit the groundings of the sentences, though we do not use these directly here.

4.3 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 learned, a set of weights θ and the lexicon Λ. For both the parser and the associated language-vision component, Λ is used to structure inference. To induce new lexical entries, we employ a variant of *GENLEX* (Artzi and Zettlemoyer, 2013) that takes as input a validation function — the compatibility between a parse and the video. 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 or used.

The joint model must learn these parameters despite three sources of noise. First, the vision-language component may simply fail to produce the correct likelihood because machine vision is far from perfect. Overcoming this requires large beam widths to avoid falling into local minima due to these errors.

Second, an infinite number of possibly-erroneous parses are true of a 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 are true and cannot be distinguished from the ground-truth parse — which is not available — by the vision component. This is a far less constrained environment than other approaches to semantic parsing. It is easy to be misguided by a loss function that is often true when it should not be and thus create many special-purpose definitions of words that happen to fit the peculiarities of any video. This results in two different problems: assigning empty semantics to many words since the likelihood of a subset of a parse is always the same or higher than the whole parse and excessive polysemy where the meaning of a word is highly specific to some irrelevant feature in a video. We introduce two features to the parser that bias it against empty semantics and against excessive polysemy. Models of communication such as the Rational Speech Acts model (Frank and Goodman, 2012) predict that speakers will avoid inserting meaningless words. One feature counts the number of predicates mapped onto semantic forms which are empty that occur in each parse. The other feature attempts to prevent excessive polysemy by counting how many new semantic forms are introduced for existing tokens by the generated entries from each parse. As the parser becomes more capable of handling sentences in the training set, these features begin to bias it against adding empty semantics and new semantic forms.

Third, models in computer vision are computationally expensive while many evaluations of parse-video pairs are required to train a parser. To overcome this, we construct a provably-correct cache that keeps track of failing subexpressions. This is possible because of a feature of this particular vision-language scoring function: the score decreases monotonically with the number of constraints. With these improvements, the modified semantic parser employing vision-language-based validation learns to map sentences into semantic parses despite facing a challenging setting with few examples and much ambiguity.

5 Dataset

We collected and annotated a dataset of captioned videos with fully annotated semantic parses of the captions. The videos contain people carrying out one of 15 actions, such as picking things up and putting things down, with one of 20 objects spanning 10 different colors. We control for 11 spatial relations between objects and actors. Many 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.

On Mechanical Turk we asked participants to provide sentences that describe something about the video. We did not specify what participants should describe to avoid biasing them and to add richness to the dataset. This sometimes led to sentences that referred to properties of the video that are well beyond the capacities of the vision system, e.g., descriptions of an agent being lazy or references to the camera's movement. We removed such sentences. At training time, the parser receives captioned videos but no annotations about which objects those captions refer to. Each sentence was annotated with a ground-truth semantic form by two trained annotators using a set of 34 predicates. Each sentence was then reviewed and corrected by one other annotator.

To detect the objects in the videos, we used two off-the-shelf detectors, OpenPose (Cao et al., 2017) for person detection and YOLO version 3 (Redmon and Farhadi, 2018) for the remaining objects. In each case we significantly lowered the confidence threshold to avoid false negatives. Many objects in this dataset are small and are handled by humans, which leads to regular object detector failures that are only partially compensated for by lowering the detection threshold at the cost of a large number of false positives. We rely on the inference mechanism of the grounded parser to automatically eliminate these numerous false positives as candidates when grounding sentences due to their low likelihoods. False negatives are much more misleading and difficult to overcome than false positives. It is harder to read in where an unseen object might be than to eliminate a low-confidence detection.

In total, the dataset contains 1200 captions from 401 videos, which selected out of a larger body of sentences collected and pruned as described above. This is comparable to the size of other datasets used for semantic parsing such as two datasets from Tang and Mooney (2001) with 880 and 640 examples respectively and the navigation instruction dataset (Chen and Mooney, 2011) with 706 examples (containing 3236 single sentences). The sentences comprising our dataset contain 169 unique tokens with an average of 7.93 tokens per caption. There are an average of 2.31 objects per caption.

6 Evaluation

6.1 Experimental Setup

We adapted the Cornell SFP (Semantic Parsing Framework) developed by Artzi (2016) to jointly reason about sentences and videos. We selected 720 examples for training and used 120 examples for the validation set to fine-tune the model parameters. We used the remaining 360 examples for the test set. This split was fixed and used in all experiments below. No sentences or videos occurred in both the training and test sets. During training, each hypothesized parse for each sentence is marked as either correct or incorrect, using either direct supervision with the target parse or compatibility with the video, depending on the experiment.

We use beams of 80 for the CKY-parser and *GENLEX*. CCG-based semantic parsers are seeded with a small number of generic combinations of syntactic and semantic types. For example, Artzi (2016) seed with 141 lexical entries; we provide 98. *GENLEX* uses these entries along with an ontology to form new syntactic and semantic types.

Precision		Recall		F1	
Direct supervision					
0.851	*0.946*	0.84	*0.933*	0.846	*0.939*
Noisy supervision (60%)					
0.235	*0.423*	0.201	*0.362*	0.217	*0.390*
Shuffled labels (direct supervision)					
0.147	*0.384*	0.122	*0.321*	0.136	*0.349*
Shuffled videos (weak supervision)					
0.000	*0.106*	0.000	*0.103*	0.000	*0.104*
Object-only vision					
0.051	*0.387*	0.042	*0.349*	0.046	*0.367*
Vision-language					
0.223	*0.663*	0.183	*0.553*	0.201	*0.591*

Figure 3: Pairs of results for each condition. On the left, we show exact match results and on the right, in *italics*, results for the near miss metric. In the case of *direct supervision*, we train with the target parses. In the case of *noisy supervision*, a percentage of the time (60% here) the parser randomly accepts or rejects a parse. In the case of *shuffled labels*, the target logical forms are assigned to random sentences. For *shuffled videos* the sentences are assigned to random videos. The likelihood of any sentence being true of a random video is low. In the case of *object-only vision*, the vision system consists solely of an object detector discarding any other predicates. The full *vision-language* approach learns to parse a significant fraction of sentences, far outperforming the object-only approach, and usually being within one predicate of the correct answer.

6.2 Results

Figures 3 and 4 summarize the experiments and ablation studies performed. The metrics we use when reporting results are *exact* matches, where the predicted parses must perfectly match the target parses, and *near misses*, where a single predicate in the semantic parse is allowed to differ from the target. Experiments were averaged across 5 runs.

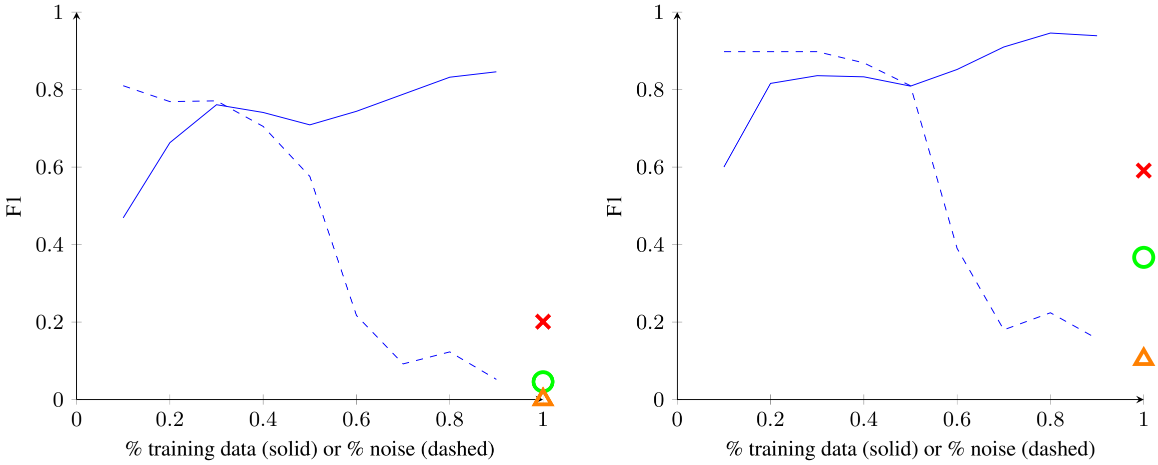
To establish chance-level performance, we trained the directly supervised approach on shuffled labels, assigning random correct parses to random sentences. This is more powerful than a simple chance-level performance calculation as the parser can still take advantage of any dataset biases. Even with the ability to exploit potential biases, performance is very low with F1 scores of 0.136 and 0.349 for the exact and near miss metrics. Both metrics pose a challenging learning problem.

As a baseline, we directly supervised the parser with the target logical forms. When doing so, it achieved high performance with F1 scores of 0.841 and 0.911 for the exact match and near miss cases. Figure 4 shows performance of direct supervision as a function of training set size.

We then added noise to the directly supervised parser. Doing so simulates the unreliable nature of vision and, to an extent, the ambiguities inherent in vision. Noise was introduced by modifying the compatibility function which determines if a parse is correct. A certain percentage of the time, that 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 noisy and near miss cases. Figure 4 shows performance of the noisy baseline as a function of how much noise was introduced.

The fully grounded parser produced 0.2 and 0.6 F1 scores for the exact and near miss metrics. This is far beyond 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 hypothesized parse may carry the same meaning as the target logical form yet it will be considered incorrect. This is less of a problem with direct supervision where the preferences that annotators have for a particular way of encoding the meaning of a sentence can be learned. In the grounded case, this cannot be learned; visually equivalent parses are equally likely. Second, computer vision is unreliable, i.e., object detectors fail. We find that in many of our videos while person detection is fairly reliable, object detection is unreliable. 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 5 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 visual correlations, like the presence or absence of particular objects, as opposed to 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, near-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 to infer about agents and interactions is crucial for learning.



(a) Exact match

(b) Near miss

Figure 4: Results from training the grounded semantic parser. 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 object detector ablation. The orange triangle represents *shuffled videos* and shows chance performance. While direct supervision outperforms vision-only supervision, the grounded parser closes the gap and operates like noisy direct supervision with roughly 55% noise.

7 Discussion

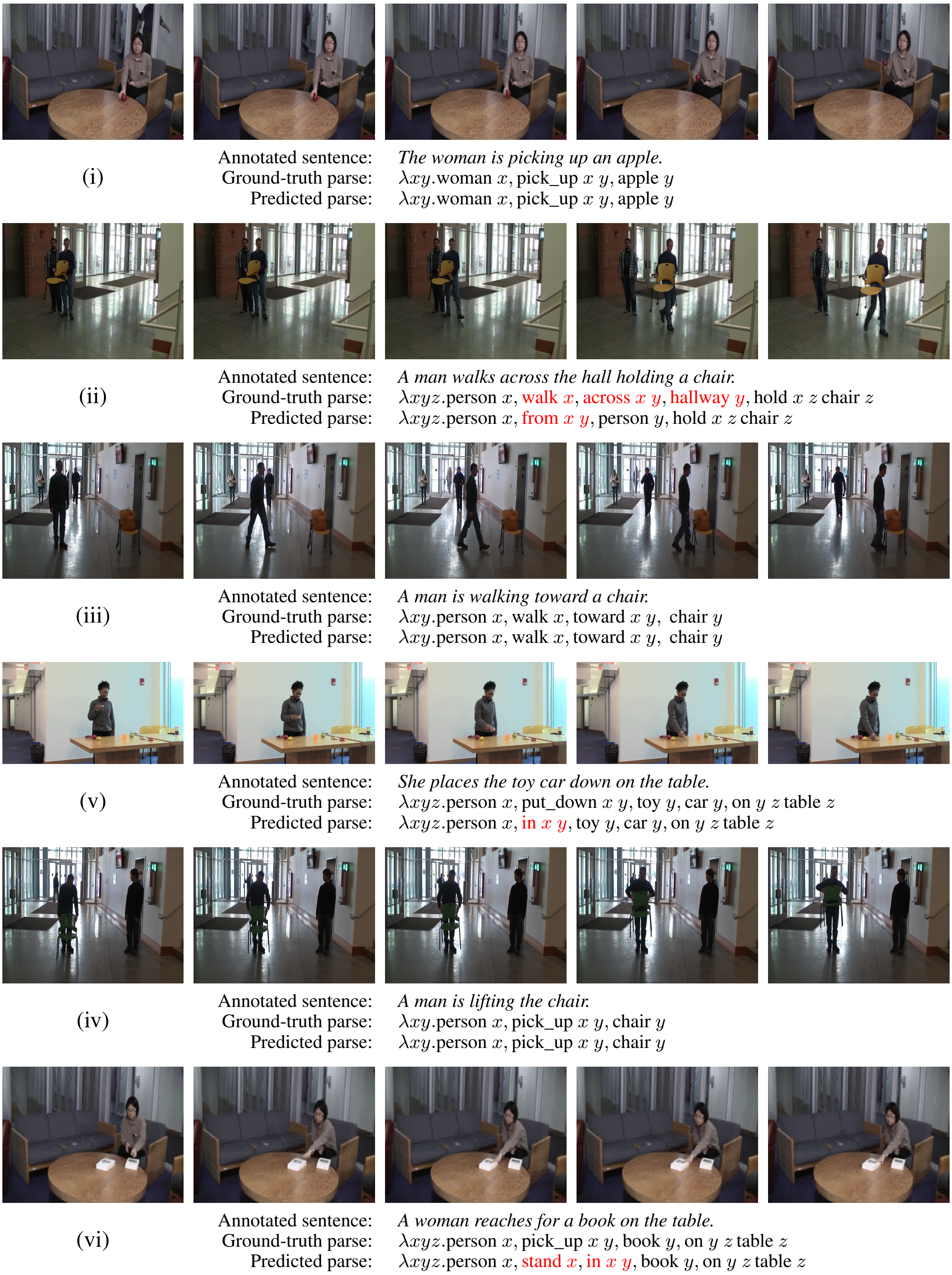
We present a semantic parser that learns the structure of language using weak supervision from vision. At test time, the model parses sentences without the need for visual input. Learning by passive observation in this way extends the capabilities of semantic parsers and points the way to a more cognitively plausible model of language acquisition. Several limits remain. Evaluating parses as correct or incorrect depending on a match to a human-annotated logical form is an overly strict criterion and is a problem that also plagues fully-supervised syntactic parsing (Berzak et al., 2016). Since two logical forms may express the same meaning, it is not yet clear what an effective evaluation metric is for these grounded scenarios. In addition, learning in such a passive scenario is hard as correlations between events, e.g., every *pick up* event involves a *touch* event, are very difficult to disentangle.

An interesting source of error in the experimental results comes from visual ambiguities. At the level of relative motions of labeled bounding boxes, the analysis performed by the language-vision system we employed here has difficulty distinguishing certain parts of actions. For example, carrying a shirt and wearing a shirt appear very similar to one another as they are actions that mostly involve moving alongside a person detection. Moreover, since every agent is wearing a shirt it becomes more difficult to learn to distinguish the two actions using positive evidence alone, i.e., a maximum likelihood approach. A more robust vision system, perhaps including object segmentations, person pose, and weak negative evidence for the occurrence of actions, would likely significantly improve the results presented.

In the future, we intend to add a generative model along with a physical simulation allowing the learner to imagine scenarios where a predicate might not hold. This would help mitigate systematic correlations between sentences and videos. 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 intend to combine the weakly supervised parser with an unsupervised parser and learn to determine whether a sentence should be grounded visually during training. We hope 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 users' homes.

Acknowledgments

This work was supported, in part, by the Center for Brains, Minds and Machines, NSF STC award 1231216, NSF Graduate Research Fellowship, Ford Foundation Graduate Research Fellowship, Toyota Research Institute, AFRL contract No. FA8750-15-C-0010, and MIT-IBM Brain-Inspired Multimedia Comprehension project.



(i)

Annotated sentence: *The woman is picking up an apple.*
Ground-truth parse: $\lambda xy.$woman x, pick_up $x\ y$, apple y
Predicted parse: $\lambda xy.$woman x, pick_up $x\ y$, apple y

(ii)

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

(iii)

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

(v)

Annotated sentence: *She places the toy car down on the table.*
Ground-truth parse: $\lambda xyz.$person x, put_down $x\ y$, toy y, car y, on $y\ z$ table z
Predicted parse: $\lambda xyz.$person x, in $x\ y$, toy y, car y, on $y\ z$ table z

(iv)

Annotated sentence: *A man is lifting the chair.*
Ground-truth parse: $\lambda xy.$person x, pick_up $x\ y$, chair y
Predicted parse: $\lambda xy.$person x, pick_up $x\ y$, chair y

(vi)

Annotated sentence: *A woman reaches for a book on the table.*
Ground-truth parse: $\lambda xyz.$person x, pick_up $x\ y$, book y, on $y\ z$ table z
Predicted parse: $\lambda xyz.$person x, stand x, in $x\ y$, book y, on $y\ z$ table z

Figure 5: Six examples of frames from videos in the dataset along with target and predicted logical forms showing both successes and failures. Failures are highlighted in red. Note how incorrect parses are usually similar to the correct semantic forms. The intended meaning is often preserved even in these cases.

References

Muhannad Al-Omari, Paul Duckworth, David C Hogg, and Anthony G Cohn. 2017. Natural language acquisition and grounding for embodied robotic systems. In *AAAI*. pages 4349–4356.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2425–2433.

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* pages 49–62.

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

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Annual Meeting of the Association for Computational Linguistics*.

Yevgeni Berzak, Andrei Barbu, Daniel Harari, and Boris Katz. 2015. Do You See What I Mean ? Visual Resolution of Linguistic Ambiguities. *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. *Conference on Empirical Methods in Natural Language Processing* .

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

Bob Carpenter. 1997. *Type-logical semantics*. MIT press.

David L. Chen and Raymond J. Mooney. 2011. Learning to interpret natural language navigation instructions fro mobservations. In *AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA.

James R Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, pages 33–36.

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

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojtek Czarnecki, Max Jaderberg, Denis Teplyashin, et al. 2017. Grounded language learning in a simulated 3D world. *arXiv preprint arXiv:1706.06551* .

Thomas Kollar, Jayant Krishnamurthy, and Grant P Strimel. 2013. Toward interactive grounded language acqusition. In *Robotics: Science and systems*. volume 1, pages 721–732.

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

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

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*.

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

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.

Lappoon R. Tang and Raymond J. Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *European Conference on Machine Learning*. pages 466–477.

Sida I Wang, Percy Liang, and Christopher D Manning. 2016. Learning language games through interaction. *Meeting of the Association for Computational Linguistics* .

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 categorial grammars. In *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. AUAI Press, Arlington, Virginia, pages 658–666.