Learning to Make Inferences in a Semantic Parsing Task

Kyle Richardson and **Jonas Kuhn**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
{kyle,jonas.kuhn}@ims.uni-stuttgart.de

Abstract

We introduce a new approach to training a semantic parser that uses textual entailment judgements as supervision. These judgements are based on high-level inferences about whether the meaning of one sentence follows from another. When applied to an existing semantic parsing task, they prove to be a useful tool for revealing semantic distinctions and background knowledge not captured in the target representations. This information is used to improve the quality of the semantic representations being learned and to acquire generic knowledge for reasoning. Experiments are done on the benchmark Sportscaster corpus (Chen and Mooney, 2008), and a novel RTE-inspired inference dataset is introduced. On this new dataset our method strongly outperforms several strong baselines. Separately, we obtain state-of-the-art results on the original Sportscaster semantic parsing task.

1 Introduction

Semantic Parsing is the task of automatically translating natural language text to formal meaning representations (e.g., statements in a formal logic). Recent work has centered around learning such translations using parallel data, or raw collections of text-meaning pairs, often by employing methods from statistical machine translation (Wong and Mooney, 2006; Jones et al., 2012; Andreas et al., 2013) and parsing (Zettlemoyer and Collins, 2009; Kwiatkowski et al., 2010). Earlier attempts focused on learning to map natural language questions to simple database queries for database retrieval using collections of target questions and formal queries. A more recent focus has been on learning representations using weaker forms of supervision that require minimal amounts of manual annotation effort (Clarke et al., 2010; Liang et al., 2011; Krishnamurthy and Mitchell, 2012; Artzi and Zettlemoyer, 2013; Berant et al., 2013; Kushman et al., 2014).

For example, Liang et al. (2011) train a semantic parser in a question-answering domain using the *denotation* (or answer) of each question as the sole supervision. Particularly impressive is their system's ability to learn complex linguistic structure not handled by earlier methods that use more direct supervision. Similarly, Artzi and Zettlemoyer (2013) train a parser that generates higher-order logical representations in a navigation domain using low-level navigation cues. What is missing in such approaches, however, is an explicit account of entailment (e.g., learning entailment rules from such corpora), which has long been considered one of the *basic aims* of semantics (Montague, 1970). An adequate semantic parser that captures the core aspects of natural language meaning should support inferences about sentence-level entailments (i.e., determining whether the meaning of one sentence follows from another). In many cases, the target representations being learned remain inexpressive, making it difficult to learn the types of semantic generalizations and world-knowledge needed for modeling entailment (see discussion in Schubert (2015)).

Attempts to integrate more general knowledge into semantic parsing pipelines have often involved additional hand-engineering or external lexical resources (Wang et al., 2014; Tian et al., 2014; Beltagy et al., 2014). We propose a different learning-based approach that uses textual inference judgements between sentences as additional supervision to learn semantic generaliza-

Transactions of the Association for Computational Linguistics, vol. 4, pp. 155–168, 2016. Action Editor: Mark Steedman.
Submission batch: 10/2015; Revision batch: 2/2016; Published 5/2016.

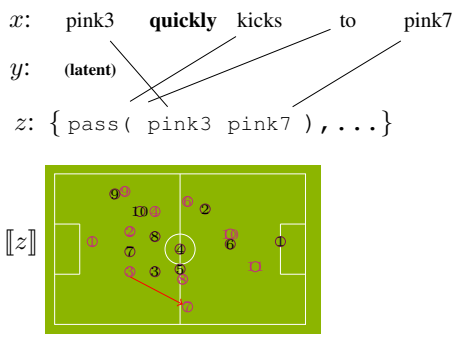


Figure 1: The original Sportscaster training setup: a text x paired with a *set* of meaning representations z derived from events occurring in a 2-d soccer simulator. The goal is to learn a latent translation, y, from the text to the correct representation.

	Text t	Hypothesis h	Entailment $t \to h$ / $h \to t$	Naïve
1.	Pink 3 quickly kicks to pink 7 `pass(pink3,pink7)`	Pink 3 kicks over to pink 7 `pass(pink3,pink7)`	`Entail` `Unknown`	`Entail`
2.	Purple 10 kicks the ball `kick(purple10)`	Purple 10 shoots for the goal `kick(purple10)`	`Unknown` `Entail`	`Entail`
3.	Pink 10 kicks the ball `kick(pink10)`	Pink 10 passes over to pink 7 `pass(pink10,pink7)`	`Unknown` `Entail`	`Contr.`
4.	Purple 7 makes a long kick `kick(purple7)`	Purple team scores another goal `playmode(goal_l)`	`Unknown` `Unknown`	`Contr.`

Figure 2: Example sentence pairs and semantic representations with textual inference judgements. *Naïve* entailments are a type of close-world assumption that result from matching semantic representations and assigning an entailment for matches and a contradiction otherwise.

tions in a semantic parsing task. Our assumption is that differences in sentence realizations provide a strong, albeit indirect, signal about differences in meaning. When paired with entailment judgements, this evidence can reveal important semantic distinctions (e.g., sense distinctions, modification) that are not captured in target meaning representations. These judgements can also be used to learn general knowledge about a domain (e.g., meaning postulates or ontological relations).

In this paper, we introduce a novel recognizing textual entailment (RTE) inspired inference task for training and evaluating semantic parsers that extends previous approaches. Our method learns jointly using structured meaning representations (as done in previous approaches) and raw textual inference judgements as the main supervision. In order to learn and model entailment phenomena, we introduce a new method that integrates natural logic (symbolic) reasoning (MacCartney and Manning, 2009) directly into a data-driven semantic parsing model.

We perform experiments on the Sportscaster corpus (Chen and Mooney, 2008), which we extend by annotating pairs of sentences in the original dataset with inference judgements. On a new inference task based on this extended dataset, we achieve an accuracy of 73%, which is an improvement of 13 percentage points over a strong baseline. As a separate result, part of our approach outperforms previously published results (from around 89% accuracy to 96%) on the original Sportscaster semantic parsing task.

2 Motivation

In this section, we describe the idea of modeling inference in a semantic parsing task using examples from the Sportscaster domain.

2.1 Problems of Representation

Figure 1 shows a training example from the original Sportscaster corpus used in Chen and Mooney (2008), consisting of a text x paired with a set of formal meaning representations z. The goal for training a semantic parser in this setup is to learn a hidden translation y from the text to the correct representation using such raw pairs as supervision. In this case, human commentary (i.e., x) was collected by having participants watch a 2-d simulation of several Robocup[1] soccer league games and comment on events in the game.

Rather than hand annotating the verbal sports commentary, sentences were paired with symbolic (logical) representations underlying the original simulator actions (André et al., 2000). These representations serve as a proxy for the grounded game context and the denotation of individual events (shown as $[\![z]\!]$). While the representations capture the general events being discussed, they often fail to capture other aspects of meaning and additional details that the human commentators found to be relevant and expressed verbally.

These issues are illustrated in Figure 2, where

[1] http://www.robocup.org/

example sentences are shown with target meaning representations. Sentence-level entailment judgements[2] between orderings of text are shown using a standard 3-way entailment scheme (Cooper et al., 1996; Bentivogli et al., 2011), along with a *naïve* inference computed by comparing the target labels. The mismatch between some of the naïve inferences and the actual entailment judgements show that the target representations alone fail to capture certain semantic distinctions. This is related to two problems:

Imprecise labels: The corpus representations fail to account for certain aspects of meaning. For example, the first two sentences in Figure 2 map to the same formal meaning representation (i.e., `pass(pink3,pink7)`) despite having slightly different semantics and divergent entailment patterns. This shift in meaning is related to the adverbial modifier *quickly*, which is not explicitly analyzed in the target representation. The same is true for the modifier *long* in example 4, and for all other forms of modification. For a semantic parser or generator trained on this data, both sentences are treated as having an identical meaning.

As shown in the example 2, other representations fail to capture important sense distinctions, such as the difference between the two senses of the `kick` relation. While *shooting for the goal* in general entails *kicking*, such an entailment does not hold in the reverse direction. Without making this distinction explicit at the representation level, such inferences and distinctions cannot be made.

Missing Domain Knowledge: Since the logical representations are not based on an underlying logical theory or domain ontology, semantic relations between different symbols are not known. For example, computing the entailments in example 3 requires knowing that in general, a `pass` event entails or implies a `kick` event (i.e., the set of things *kicking* at a given moment includes the set of people *passing*). Other such factoids are involved in reasoning about the sentences in example 4: `purple7` is part of the `purple team`, and a `score` event entails a `kick` event (but not conversely).

Our goal is to learn a semantic parser that can capture the inferential properties of language described above. Rather than re-annotating the corpus and creating a domain ontology from scratch, we use the raw entailment judgements to help improve and learn about the existing representations. We show that entailment judgements prove to be a powerful tool for solving the two problems described above.

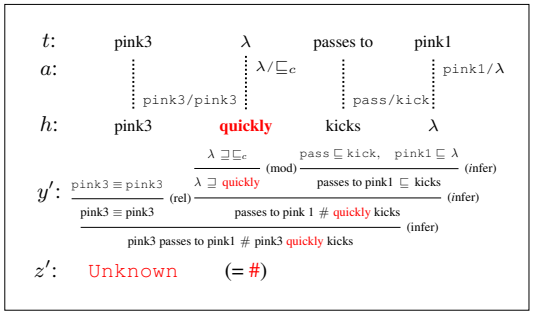


Figure 3: The inference training setup: an ordered pair (t, h) annotated with a sentence-level inference relation z'. The goal is to learn a hidden alignment a between t, h, and a hidden proof (tree) y' that generates the target inference.

2.2 Learning from Entailment

Our approach addresses the problems outlined above by adding pairs of text annotated with inference judgements (as shown in Figure 2) to the original Sportscaster data (as shown in Figure 1). While training an ordinary semantic parser, we use such pairs to jointly reason about the Sportscaster concepts/symbols and prove theorems about the target entailments using a simple logical calculus. The idea is that these proofs reveal distinctions not captured in the original representations, and can be used to improve the semantic parser's internal representations and acquire knowledge.

This training setup is illustrated in Figure 3, where each training instance consists of a text t and hypothesis h, and an entailment judgement z'. The goal is to learn a hidden proof y' that derives the target entailment by transforming the text into the hypothesis. Such a proof is driven by latent semantic relationships (shown on the top row in y' and *rel*) between aligned pairs of symbols (the arc labels in a, delimited by "/"). These relations record the effect of substituting or inserting/deleting symbols in the text with related symbols in the hypothesis and compare the denotations of these symbols. These relations are then projected up a proof tree using generic inference rules

[2]We adopt the definition of *entailment* used in the RTE challenges (Dagan et al., 2005): a text T entails a hypothesis H if "typically, a human reading T would infer that H is most likely True"

(*infer* and *mod*) to compute a global inference.

This proof gives rise to several new facts: the `pass` symbol is found to forward entail or imply (shown using the set inclusion symbol $\sqsubseteq$) the `kick` symbol. The adverbial modifier, which is previously unanalyzed, is treated as an entailing modifier $\sqsubseteq_c$, which results in a reverse entailment or implication (shown using the symbol $\sqsupseteq$) when inserted (or substituted for the empty symbol λ) on the hypothesis side. The first fact can be used for building a domain theory, and the second for assigning more precise labels to modifiers for the semantic parser. The overall effect of inserting the adverbial modifier (shown in red) is then propagated up the proof tree leading to an `Uncertain` inference (shown using the # symbol).

Computing entailments is driven by learning the correct semantic relations between primitive domain symbols, as well as the semantic effect of deleting/inserting symbols. We focus on learning the following very broad types of linguistic inferences (Fyodorov et al., 2003): **construction-based** inferences, or inferences generated from specific (syntactic) constructions or lexical items in the language, and **lexical-based** inferences, or inferences generated between words or primitive concepts due to their inherent lexical meaning.

Construction-based inferences are inferences related to modifier constructions: *quickly* (`pass`) $\sqsubseteq$ `pass`, `goal` $\sqsupseteq$ *nice* (`goal`), *gets_a* (`free kick`) $\equiv$ (equivalence) `free kick`, where the entailments relate to default properties of particular modifiers when they are added or dropped. Lexical-based inferences relate to general inferences and implications between primitive semantic symbols or concepts: `kick` $\sqsupseteq$ `score`, `pass` $\sqsubseteq$ `kick`, and `pink1` $\sqsubseteq$ `pink team`.

2.3 Outline of Approach

Experiments are done by first training a standard semantic parser on the Sportscaster dataset, then improving this parser using an extended corpus of sentences annotated with entailment judgements. Semantic parsing is done using a probabilistic grammar induction approach (Börschinger et al., 2011; Angeli et al., 2012), which we extend to accommodate entailment modeling. The natural logic calculus is used as the underlying logical inference engine (MacCartney and Manning, 2009).

To evaluate the quality of our resulting semantic parser and the acquired knowledge, we run our system on a held-out set of inference pairs. The results are compared to the *naïve* inferences computed by the initial semantic parser.

3 Semantic Parsing

In this section, we describe the technical details behind the semantic parsing. We also describe the underlying natural logic inference engine used for computing inferences, and how to integrate this into a standard semantic parsing pipeline for modeling our extended corpus.

3.1 Base Semantic Grammars

Semantic grammars (Allen, 1987) are used to perform the translation between text and logical meaning representations. The rules in these grammars are automatically constructed from the target corpus representations using a small set of rule templates, building on Börschinger et al. (2011) (henceforth BJJ).

Rule Templates and Extraction Figure 4 shows a set of rule templates in the form of context-free productions, along with examples from the Sportscaster domain. Meanings representations (MR) are atomic formulae of predicate logic and take the following form: $\texttt{Rel(x}_{\texttt{arg1}}\texttt{,..,x}_{\texttt{argN}}\texttt{)}$. The production rules break down each representation to smaller parts: **lexical** rules associate MR constituents or symbols (e.g., $\texttt{Rel,x}_{\texttt{arg1}}$ instances) to individual words, **phrase** rules associate these constituents to word sequences, **concept** rules associate phrase rules to domain concepts, and **glue** rules combine concepts to build complete MRs.[3]

Lexical rules are created by breaking down all MRs in a target corpus, and associating each constituent with all words in the target corpus. Phrase rules are from (Johnson et al., 2010) and allow arbitrary word sequences to be associated with constituents as opposed to single words. Such rules can be used to **skip** words that don't contribute direct meaning to a constituent or are unanalyzed, which is represented using the empty word symbol λ_{w}. This is shown in the treatment of the adverbial *quickly* in the phrase *passes quickly to* in Figure 4.2a.

Glue rules are constructed by marking constituent concepts with syntax-semantic roles and

[3]Capital letters (e.g., $E, A, ..$) are used as variables in the grammar to refer to sets of symbol types, and x is used to refer to all symbols in the grammar.

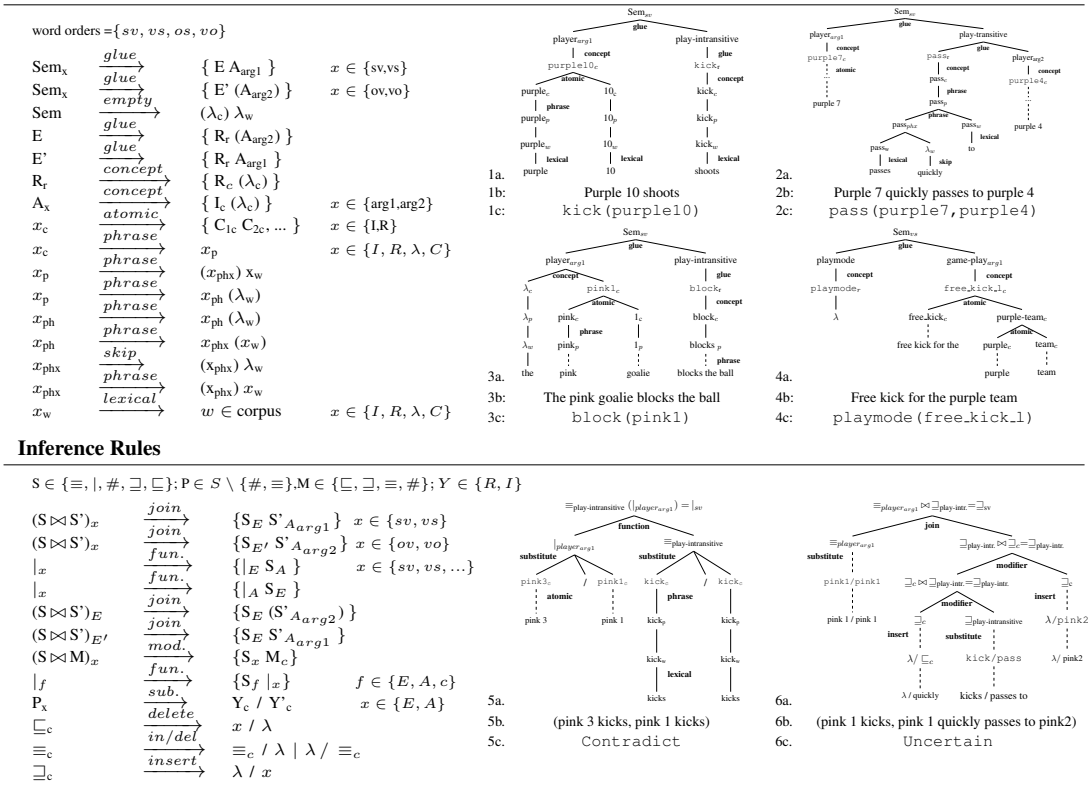


Figure 4: The top are rule templates for building a semantic grammar with examples from the Sportscaster domain. The bottom are templates for encoding natural logic inference rules as grammar rules. Rules in $\{.\}$ are expanded to all orders. Example derivations are shown on the right (some derivations are collapsed using dashed lines).

combining these roles according to the structure of the MRs. For example, the constituent symbol `block` in Figure 4.3 is marked as an *intransitive-play* relation and `pink1` as a *play-argument*, both of which combine to create a well-formed MR. Here we diverge from BJJ, where such abstractions are not used and full MRs are encoded as separate grammar symbols. As in BJJ, word-order rules are used to account for regularities in the order in which arguments combine with relations.

λ_c and Atomic Rules In addition to the skip phrase rules, concept rules are padded with a new empty concept λ_c, which are used for modeling phrases and modifiers surrounding concept phrases. For example, in *The pink goalie* in Figure 4.3a, *the* is treated as a separate phrase that modifies *pink goalie*. As described later, these phrases will get classified according to their effect on entailment using our extended corpus, but in the base semantic grammar get treated as not contributing any additional meaning.

The **atomic** rules optionally break down some of the domain symbols to smaller concepts, rather than using the original corpus symbols directly as in BJJ. For example, the concept `pink3` is treated as consisting of two concepts: `pink` and `3`. Similarly, the game symbol `free_kick_l` is broken down to two concepts: `free kick` and `purple team` (or *l*). Unlike in BJJ, some flexibility is permitted in terms of dropping or skipping over some constituent symbols that do not get realized in sentences. For example, `playmode` in Figure 4.4a is dropped, since it is not explicitly described in the associated text.

Interpretation Using this grammar, a given sentence input will generate a large space of output derivations, each related to a particular semantic representation. An interpretation of a derivation d is the MR produced from the derivation by applying the glue rules. By assigning probabilities to

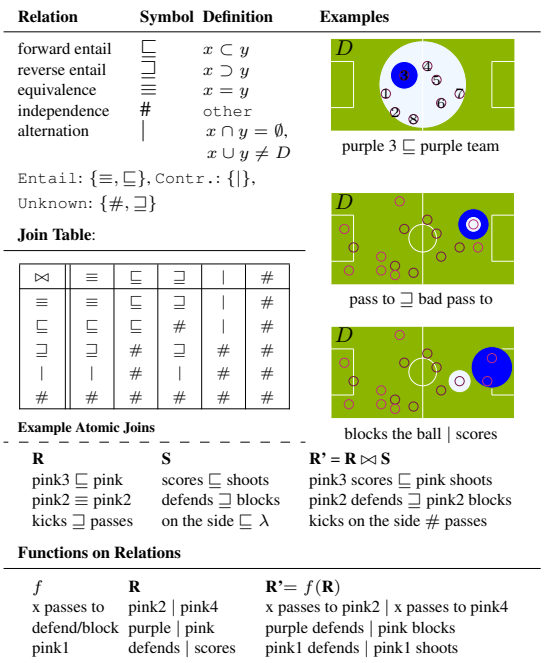


Figure 5: Components of the natural logic inference system. The top table shows primitive set-theoretic inference relations with sports examples (right), the middle table defines the join function used for combining these relations, bottom table show examples of functions on these relations.

the rules in our grammar, we can learn the correct interpretations using our training data.

3.2 Entailment Modeling

The base semantic parser described in the previous section makes it possible to translate sentences to formal representations. Entailment modeling aims to discover abstract relations between symbols in these representations. For example, knowing how the meaning, or *denotation*, of the symbol `purple7` in general relates to the meaning of `purple team`, or how `score` relates to `kick`.

In this section, we describe our general framework used for modeling textual entailment.

Natural Logic Calculus

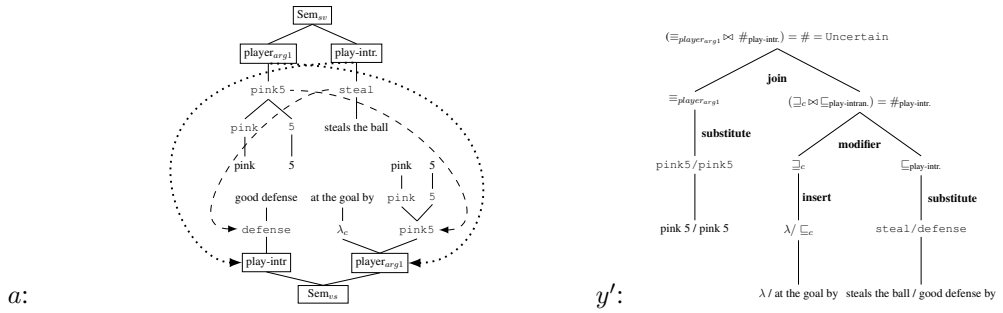
We use a fragment of the natural logic calculus to model entailment (MacCartney and Manning, 2009; Icard III, 2012). Natural logic derives from work in linguistics on proof-theoretic approaches to semantics (van Benthem, 2008; Moss, 2010). More recently, it has been used in NLP for work on RTE (MacCartney and Manning, 2008; Angeli and Manning, 2014; Bowman et al., 2014).

Components of the calculus are shown in Figure 5. A small set of primitive set-theoretic relations are defined, which are used to relate the denotations of arbitrary lexical items (w.r.t to a domain of discourse D). We use a subset of the original seven relations to relate symbols (and by extension, word/phrases) in our domain. For example, `purple3` (or "purple 3") has a ⊑ (or subset) relation to `purple team` (or "purple team"), which is illustrated in Figure 5 using a Venn diagram. These primitive relations are then composed using two operations: atomic join rules, or generic inference rules for combining two inference relations to create a new relation (shown in the join table), and function rules, or inference rules associated with particular lexical items that project certain properties onto other relations.

Sentence-level entailment recognition is done by finding an alignment between a text and hypothesis pair. Such an alignment transforms the text into the hypothesis by substituting each part of the text with lexical items in the hypothesis, and inserting/deleting other items. Each local transformation is marked with a semantic relation, and these relations are composed using the join and function rules. A proof tree records the result of this overall process, and the top-most node shows the overall inference relation (e.g., | in Figure 4.5a or ⊒ in Figure 4.6a).

Semantic relations between symbols and functions are usually recorded in a semantic lexicon. Since we have no prior knowledge about how symbols relate to one another, we learn these relations using the resulting entailment judgements as supervision. For example, since we do not know the exact relation between `kick` and `pass`, we start by assuming all semantic relations and find the correct relation by looking into the (latent) proof trees that produce the correct entailments in our training data (e.g., the tree in Figure 3).

All actions in our domain are fixed in time and place, and as such apply to a unique group of objects. For example, if a passing event is being described, the person doing the passing or being passed to at a particular moment must always be a unique individual. Substituting this individual with someone else will always result in a contradiction (see examples on the bottom of Figure 5). We therefore use a single default function rule that always projects negations | up the proof tree. See (MacCartney, 2009) for more information about



$((t = \text{“pink 5 steals the ball”}, h = \text{“good defense at the goal by pink 5”}), z' = \texttt{Uncertain})$

Figure 6: An example produced by our model: (t, h) are the text and hypothesis, z' is the inference annotation/relation that holds between $t \rightarrow h$, a is a phrase alignment between both sentences, and y' is a (simplified) proof tree that generates the target inference.

these types of *functional relations*.

Inference Grammar Rules

We encode natural logic operations as production rules and add them to the base semantic grammar described in Section 3.1. Rule templates are shown on the bottom of Figure 4. **Substitute** rules assign a semantic relation to a pair of symbols: e.g., $\sqsupseteq_{\text{play-intr.}} \rightarrow \texttt{kick}_c$ / $\texttt{pass}_c$, where the subscript on the semantic relation is the role of the left concept being substituted. Substitutions occur between symbols with the same role, such as all relation symbols or all argument symbols in a domain and set of MRs. **Function** rules project negations (regardless of role) up a proof tree. **Join** rules compose relations using the join function $\bowtie$ and, like the glue rules, are used to construct well-formed MRs.

Substitution rules arbitrarily assign semantic relations to pairs of symbols since the correct relation is not known at the start (as discussed previously). This set of relations can be constrained by adding knowledge into the grammar. In our experiments, we assume a single negation rule between all arguments of the same semantic type (e.g., *player* arguments).

Modifiers and Senses We add two concept symbols to the grammar: $\sqsubseteq_c$ and $\equiv_c$ to replace the λ_c/λ_w rules in the base grammar. These are used to classify modifiers (e.g., the adverbial modifier in Figure 2.1) and other expressions that are unanalyzed. The **modifier** rule allows these to combine with other symbols to affect entailment in various ways when added/dropped. With the empty symbol λ, other symbols can also be arbitrarily added/dropped via the **insert** and **delete** rules.

We handle sense distinctions by allowing relation symbols (e.g., $\texttt{kick}$) in the base grammar to break down into a fixed number of specific senses in the grammar (e.g., $\texttt{kick}_1$, $\texttt{kick}_2$, ...). Using the substitution rule, these different senses can be compared in the standard way to account for the semantic distinctions discussed in Section 2.1. In our experiments, we assigned a random number of senses to the most frequent events.

Since the inference grammar is built on top of the base semantic grammar, these additional sense and modifier distinctions can be used for improving the base parser. Figure 8 shows examples of improvements in the semantic parse output after training with the extended corpus.

Construction vs. Lexical The distinction between construction-based and lexical-based inferences described in Section 2.2 is the difference between insert/delete and substitution rules in the inference grammar rules.

Interpretation A given input will generate a large set of proof trees and an even larger set of semantic relations between different symbols. The crucial aspect of the interpretation of the proof tree is the overall inference relation marked at the root node. These relations are mapped into particular inference judgements as shown in Figure 5.

Tree Alignment

The inference grammar assumes as input a word/phrase alignment between sentence pairs. Such an alignment is done in a heuristic fashion by parsing each sentence individually using the se-

mantic grammar and aligning nodes in the resulting parse trees that have matching roles. A string is produced by pairing the yield of each matching subtree using a delimiter /. Subtrees that do not have a matching role in the other tree or are modifier expressions are isolated and aligned to the empty symbol λ.

An example tree alignment is shown in Figure 6a, where the relation nodes *play-intr* and argument1 nodes $player_{arg1}$ are aligned. Since there are no modifiers in the argument subtrees, the yields of the two trees are simply combined to create the string *pink 5 / pink 5*. The modifier phrase λ_c is removed from the second relation subtree and aligned to the empty string: λ */ at the goal by*. The remaining part of the relation subtrees are then aligned: *good defense / steals the ball*. With this input, the standard phrase and concept rules from the base grammar are used to tag each phrase and inference rules are then applied to generate proofs.[4]

In our experiments, we use the tags from the semantic parse trees used during the alignment step to restrict the space of proofs considered. For example, we already know from the semantic parser output in Figure 6a that the text involves a `steal` event and the hypothesis a `defense` event, so we can constrain the search to consider only proofs that involve these two types of events.

3.3 Learning

Semantic parsing and inference computation is performed using a single generative probabilistic framework as shown in Figure 7. A probabilistic context-free grammar (PCFG) transforms input to logical representations and entailment judgements using the grammar rules defined above. Learning reduces to the problem of finding the optimal parameters θ for our PCFG given example input/output training pairs $\{(x_i, Z_i)\}_{i=1}^n$.

This is done via a EM bootstrapping approach that uses a k-best approximation of grammar derivations to estimate θ (Angeli et al., 2012). At each iteration in the EM procedure $t = 1 \dots T$, a set of k-best derivations is generated for each input x, $D(x) = \{(d_j, p_j)\}_{j=1}^k$, using the current parameters θ^t. The set of *valid* derivations,

[4]For readability, substitute rules such as $R \rightarrow X \ / \ Y$ in many of the proof trees are simplified to the following: $R \rightarrow X/Y$ and $X/Y \rightarrow$ x string/y string, without showing the full concept/phrase analysis for X, Y. See Figure 4.5a for a more precise example.

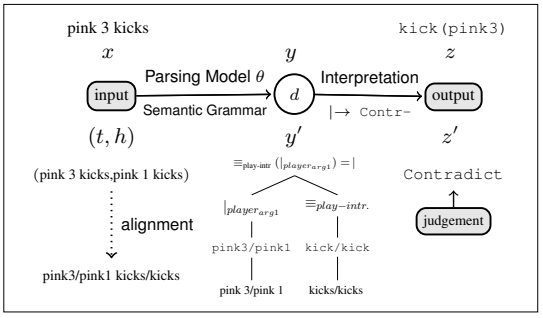


Figure 7: The prediction model: input is mapped to a hidden derivation d using a PCFG (parameterized by θ), which is then used to generate an output semantic representation. In the case of semantic parsing (top), d is a latent semantic parse tree and the output is a logical representation. For entailment detection, d is a latent proof tree and the output is a human judgement about entailment.

$C(x, Z) \subseteq D(x)$, includes all derivations that, when interpreted, are included in the set of training labels Z. From this set, $C'(x, Z)$ is computed by normalizing the probabilities, each p, to create a proper probability distribution.

For a given rule $A \rightarrow \beta$, the parameter updates are computed using these sets of normalized valid derivations. This is given by the following (unnormalized) formula (with Dirichlet prior α):

$$\theta_{A\rightarrow\beta}^{t+1} = \alpha + \sum_{i=1}^{n} \sum_{(d,p)\in C'(x_i,Z_i)} \texttt{count}(d, A \rightarrow \beta)\, p$$

4 Experiments

In this section, we discuss the Sportscaster dataset and our experimental setup.

4.1 Datasets

Sportscaster The Sportscaster corpus (Chen and Mooney, 2008) consists of 4 simulated Robocup soccer games annotated with human commentary. The English portion includes 1872 sentences paired with sets of logical meaning representations. On average, each training instance is paired with 2.3 meaning representations. The representations have 46 different types of concepts, consisting of 22 entity types and 24 event (and event-like) predicate types.

While the domain has a relatively small set of concepts and limited scope, reasoning in this domain still requires a large set of semantic relations and background knowledge. From this small set

of concepts, the inference grammar described in Section 3.2 encodes around 3,000 inference rules. Since soccer is a topic that most people are familiar with, it is also easy to get non-experts to provide judgements about entailment.

Extended Inference Corpus The extended corpus consists of 461 unaligned pairs of texts from the original Sportscaster corpus annotated with sentence-level entailment judgements. We annotated 356 pairs using local human judges an average of 2.5 times[5]. Following Dagan et al. (2005), we discarded pairs without a majority agreement, which resulted in 306 pairs (or 85% of the initial set). We also annotated an additional 155 pairs using Amazon Mechanical Turk, which were mitigated by a local annotator.

In addition to this core set of 461 entailment pairs, we separately experimented with adding unlabeled data (i.e., pairs without inference judgements) and ambiguously labelled data (i.e., pairs with multiple inference judgements) to train our inference grammars (shown in the results as *More Data*) and test the flexibility of our model. This included 250 unlabeled pairs taken from the original dataset, as well as 592 (ambiguous) pairs created by deriving new conclusions from the annotated set. This last group was constructed by exploiting the transitive nature of various inference relations and mapping pairs with matching labels in training to {`Entail`,`Unknown`}.

4.2 Training and Evaluation

We perform two types of experiments: A semantic parsing experiment (Task 1) to test our approach on the original task of generating Sportscaster representations. In addition, we introduce an inference experiment (Task 2) to test our approach on the problem of detecting entailments/contradictions between sentences.

For the semantic parsing experiment, we follow the original setup of Chen and Mooney (2008). 4-fold cross validation is employed by training on all variations of 3 games and evaluating on a left out game. Each representation produced in the evaluation phrase is considered correct if it matches exactly a gold representation.

The second experiment imitates an RTE-style evaluation and tests the quality of the background knowledge being learned using our inference grammars. Like in the semantic parsing task, we perform cross-validation on the games using both the original data and sentence pairs to jointly train our models, and evaluate on left-out sets of inference pairs. Each proof generated in the evaluation phrase is considered correct if the resulting inference label matches a gold inference.

[5]We used a version of the elicitation instructions used in the RTE experiments of Snow et al. (2008)

Task 1: Semantic Parsing	Match F_1
(Chen et al., 2010)	0.80
(Börschinger et al., 2011) (BJJ)	0.86
(Gaspers and Cimiano, 2014)	0.89
Base Grammar Only	**0.96**
Inference Grammar (IG)	0.96
IG – More Data	**0.96**

Task 2: Inference Task	Accuracy
Majority Baseline	0.33
RTE classifier	0.52
Naïve Inference	0.60
SVM Flat Classifier	0.64
IG – Lexical Inference Only	0.72
IG – Full	**0.73**
IG – More Data	0.72

Table 1: Results on the semantic parsing (top) and inference (bottom) cross validation experiments.

We implemented the learning algorithm in Section 3.3 using the k-best algorithm by Huang and Chiang (2005), with a beam size of 1,000. The base semantic grammars were each trained for 3 iterations and re-trained using the additional inference grammar rules for 10 iterations. Two Dirichlet priors were used, $\alpha_1 = 0.05$ (for lexical rules) and $\alpha_2 = 0.3$ (for non-lexical rules) throughout. Lexical rule probabilities were initialized using co-occurrence statistics estimated using an IBM Model1 word aligner (uniform initialization otherwise). 5 additional senses were added to the inference grammar for the most frequent events.

4.3 Results

The results of both tasks are shown in Table 1. Scores are averaged over all held out test sets.

Task 1: Semantic Parsing We compare the results of our base semantic parser model with previously published semantic parsing results. While our grammar model simplifies how some of the knowledge is represented in grammar derivations (e.g., in comparison to BJJ), the set of output representations or interpretations is restricted to the original Sportscaster formal representations making our results fully comparable. As shown, our base grammar strongly outperforms all previously published results.

We also show the performance of our inference grammars on the semantic parsing task after being trained with additional inference sentence pairs. This was done under two conditions: when the inference grammar was trained using fully labeled inference data and unlabeled/ambiguously labeled data (*more data*). While not fully comparable to previous results, both cases achieve the same results as the base grammar, indicating that our additional training setup does not lead to an improvement on the original task.

Task 2: Inference Task The main result of our paper is the performance of our inference grammars on the inference task. For comparison, we developed several baselines, including a majority baseline (i.e., guess the most frequent inference label from training). We also use an RTE max-entropy classifier that is trained on the raw text inference pairs to make predictions. This classifier uses a standard set of RTE features (e.g., word overlap, word entity co-occurrence/mismatch). Both of these approaches are strongly outperformed by our main inference grammar (or *IG Full*).

The *Naïve Inference* baseline compares the full Sportscaster representations generated by our semantic parser for each sentence in a pair and assigns an `Entailment` for representations that match and a `Contradiction` otherwise (see discussion in Section 2.1). This baseline compares the inferential power of the original representations (without background knowledge and more precise labels) to the inferential power of the inference grammars. The strong increase in performance suggests that important distinctions that are not captured in the original representations are indeed being captured in the inference grammars.

We tested another classification approach using a *Flat Classifier*, which is a multi-class SVM classifier that makes predictions using features from the input to the inference grammar. Such input includes both sentences in a pair, their parse trees and predicted semantic labels, and the alignment between the sentences. In Figure 6, for example, this includes all of the information excluding the proof tree in y'. This baseline aims to test the effect of using hierarchical, natural logic inference rules as opposed to a *flat* or linear representation of the input, and to see whether our model learns more than the just the presence of important words that are not modeled in the original representations. Features include the particular words/phrases aligned or inserted/deleted, the category of these words/phrases in the parse trees, the rules in both parse trees and between the trees, the types of predicates/arguments in the predicted representations and various combinations of these features. This is also strongly outperformed by our main model, suggesting that the natural logic system is learning more general inference patterns.

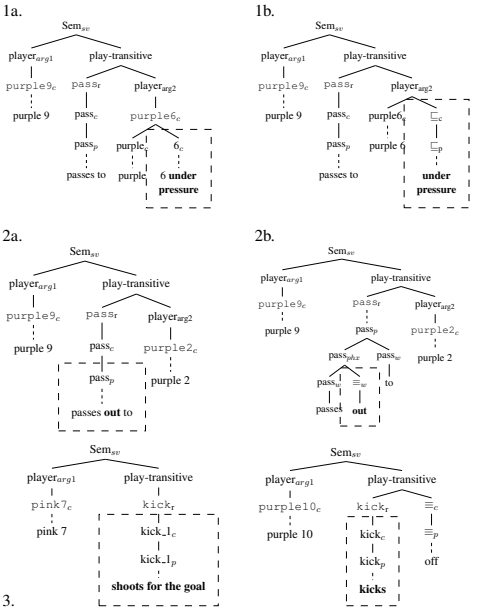


Figure 8: Example semantic parse trees (1,2) before (a) and after (b) training on the extended corpus. Example 3 shows two senses learned for the `kick` relation.

Finally, we also experimented with removing insertions and deletions of modifiers from alignment inputs to test the effect of only using lexical knowledge to solve the entailment problems (*Lexical Inference Only*). In Figure 6 this involves removing "at the goal" from the alignment input and relying only on the grammars knowledge about how `steal` (or "steals the ball") relates to `defense` (or "good defense by") to make an entailment decision. This only slightly reduced the accuracy, which suggests that the real strength of the grammar lies in its lexical knowledge.

Qualitative Analysis Figure 8 shows example parse derivations before and after being trained using the inference grammars and additional inference pairs. In example 1, the parser learns to cor-

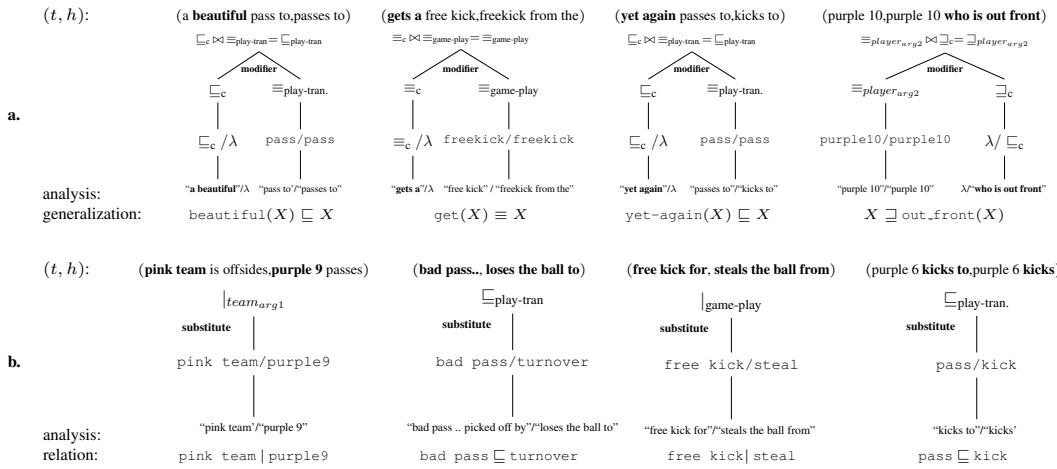


Figure 9: Example construction-based (a) and lexical-based (b) inferences (both defined in Section 2.2) taken from parts of proof trees learned during our experiments. While inferences are computed between semantic concept symbols (e.g., `purple4`, `pass`, $\sqsubseteq_c$), the *generalizations* in a. show how such structures can be used to generate lexicalized inference rules.

sense/context error:

1a. t : Pink 9 shoots
h : Pink 9 shoots for the goal
z' : `Entail` (predicted: `Uncertain`)

semantic parse error:

1b. t : Purple 8 steals the ball back
h : Purple 8 steals the ball from pink 6
z' : `Uncertain` (predicted: `Contr.`)

alignment/modifier error:

1c. t : A goal for the purple team.
h : And the purple team scored another goal
z' : `Uncertain` (predicted: `Entail`)

2.

		Predicted		
		`Entail`	`Contradict`	`Uncertain`
Gold	`Entail`	141	24	11
	`Contradict`	11	139	17
	`Uncertain`	38	22	58

Figure 10: Example inference pairs where our system fails (1a-c). A confusion matrix is shown in 2.

rectly treat the modifier "under pressure" as a separate constituent. The particular analysis also captures the correct semantics by treating this phrase as forward-entailing, which allows us to predict how the entailment changes if we insert or delete this constituent. Similarly, the parser learns a more fine-grained analysis for the phrase "passes out to" by treating "out" as a type of modifier that does not affect entailment. Example 3 shows how the improved model learns to distinguish two senses of the `kick` relation.

On the inference task, one advantage of the natural logic approach is that it is easy to see how our models make entailment decisions by looking directly at the resulting proof trees. Figure 9 shows the types of knowledge learned by our system and used in proofs. Figure 9a shows example *construction-based* inferences, or modifier constructions. For example, the first example treats the word "beautiful" in "a beautiful pass" as a type of modifier that changes the entailment or implication when it is inserted (forward-entails) or deleted (reverse-entails). In set-theoretic terms, this rule says that the set of "beautiful passes" is a subset of the set of all "passes". The model also learns the semantics of longer phrases, including how certain types of relative clauses (last example) affect entailment. Figure 9b show types of *lexical-based* inferences, or relations between specific symbols. For example, the model learns that the `pink team` is disjoint from a particular player from the purple team, `purple9`, and that a `bad pass` implies a `turnover` event.

Figure 10 shows three common cases where our system fails. The first error (1a) involves a sense error, where the system treats "shoots" as having a distinct sense from "shoots for the goal". This can be explained by observing that "shoots" is used ambiguously throughout the corpus to refer to both shooting for the goal and ordinary kicking. The second example (1b) shows how errors in the semantic parser (which is used to generate an

alignment) propagate up the processing pipeline. In this case, the semantic parser erroneously predicted that "pink 6" is the first argument of the `steal` relation (a common type of word-order error), and subsequently aligned "Purple 8" to "pink 6". Similarly, the semantic parse tree for the hypothesis in the last (1c) failed to predict "another" as a modifier, which would generate an alignment with the empty string λ.

To better understand the results, a confusion matrix for our inference grammars on the cross-validation experiments is shown in Figure 10.2. It reveals that our system is worst at predicting `Uncertain` inferences. An informal survey of a portion of the data suggests that this is largely due to the alignment/modifier errors discussed above. This is also reflected in the results that use only lexical inference rules to make predictions, which had a minimal effect on the inference performance.

5 Discussion and Conclusion

We have focused on learning representations for semantic parsing that capture the inferential properties of language. Since the goal of semantic parsing is to generate useful meaning representations, the representations being learned should facilitate entailment and inference. Since entailment is also closely tied to how we evaluate and make decisions about representations, we believe that learning methods for semantic parsing should also be able to use such judgements as supervision to influence and guide the learning. We proposed a general framework based on these ideas, which uses textual inference judgements between pairs of sentences and symbolic reasoning as a tool to learn more precise representations. While our approach uses natural logic (MacCartney and Manning, 2009) as the underlying reasoning engine, other reasoning frameworks with comparable features could be used.

Technically, our natural logic inference system is encoded as a PCFG, in which the background rules of the logic are expressed as probabilistic rewrite rules. Learning in this framework reduces to a probabilistic grammatical inference task, in this case using entailment judgements as the primary supervision. These entailments give indirect clues about a domain's denotational semantics, and can be used to reason about and find gaps in the target meaning representations. While our approach focuses on natural language, it closely relates to work on *learning from entailment* in the probabilistic logic literature (De Raedt and Kersting, 2004).

Our setup closely follows other work on situated semantic interpretation (as advocated by Mooney (2008)) and other approaches to semantic parsing that use low-level feedback to learn representations. In real situated learning tasks, however, learners will often find themselves in a situation where they observe two linguistic utterances describing the same situation. Our training setup tries to imitate these types of cases, where being able to reason and learn about entailment directly is essential.

Since capturing inference is our main goal, we also propose using textual entailment as an evaluation metric for semantic parsing. As reflected in the results, our inference task (i.e., Task 2) is considerably harder than the original semantic parsing evaluation (i.e., Task 1). This is not surprising, given that entailment recognition is in general known to involve considerable amounts of lexical and world knowledge (LoBue and Yates, 2011). Since the difference in performance on the original task is minimal between our base grammars and the inference grammars, one might conclude that the original evaluation does not tell us very much about the quality of the semantic grammar being learned in the same way as our new inference evaluation. We hope that our work pushes others in the direction of using entailment not only as a tool for learning, but for evaluating and comparing the quality of semantic parsers.

While our current model only handles simple types of inferences relating to inclusion/exclusion, we believe that our overall approach can be used to tackle more complex entailment phenomena. Future work will focus on extending our method to new datasets and inference phenomena.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) via SFB 732, project D2. We thank our anonymous reviewers and the action editor for their helpful comments and feedback. Thanks also to our IMS colleagues, in particular Christian Scheible, for providing feedback on earlier drafts, as well as to Ekaterina Ovchinnikova and Cleo Condoravdi for helpful discussions.

References

James Allen. 1987. *Natural Language Understanding*. Benjamin/Cummings Publishing Company, Inc.

Elisabeth André, Kim Binsted, Kumiko Tanaka-Ishii, Sean Luke, Gerd Herzog, and Thomas Rist. 2000. Three robocup simulation league commentator systems. *AI Magazine*, 21(1):57.

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of ACL-2013*, pages 47–52.

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of EMNLP-2014*, pages 534–545.

Gabor Angeli, Christopher D Manning, and Daniel Jurafsky. 2012. Parsing time: Learning to interpret time expressions. In *Proceedings of NAACL-2012*, pages 446–455.

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.

Islam Beltagy, Katrin Erk, and Raymond Mooney. 2014. Semantic parsing using distributional semantics and probabilistic logic. In *Proceedings of ACL-2014*, pages 7–12.

Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. 2011. The seventh Pascal Recognizing Textual Entailment challenge. *Proceedings of TAC*, 2011.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of EMNLP-2013*, pages 1533–1544.

Benjamin Börschinger, Bevan K. Jones, and Mark Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *Proceedings of EMNLP-2011*, pages 1416–1425.

Samuel R Bowman, Christopher Potts, and Christopher D Manning. 2014. Recursive neural networks can learn logical semantics. In *Proceedings of 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21.

David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: A test of grounded language acquisition. In *Proceedings of ICML-2008*, pages 128–135.

David L. Chen, Joohyun Kim, and Raymond J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *Proceedings of CoNLL-10*, pages 18–27.

Robin Cooper, Dick Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognizing Textual Entailment*, pages 1–9.

Luc De Raedt and Kristian Kersting. 2004. Probabilistic inductive logic programming. In *Algorithmic Learning Theory*, pages 19–36. Springer.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2003. Order-based inference in natural logic. *Logic Journal of IGPL*, 11(4):385–416.

Judith Gaspers and Philipp Cimiano. 2014. Learning a semantic parser from spoken utterances. In *Proceedings of IEEE-ICASSP*, pages 3201–3205.

Liang Huang and David Chiang. 2005. Better k-best parsing. In *Proceedings of IWPT-2005*, pages 53–64.

Thomas F Icard III. 2012. Inclusion and exclusion in natural language. *Studia Logica*, 100(4):705–725.

Mark Johnson, Katherine Demuth, Bevan Jones, and Michael J Black. 2010. Synergies in learning words and their referents. In *Proceedings NIPS-2010*, pages 1018–1026.

Bevan Keeley Jones, Mark Johnson, and Sharon Goldwater. 2012. Semantic parsing with Bayesian Tree Transducers. In *Proceedings of ACL-2012*, pages 488–496.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of EMNLP/CoNLL-2012*, pages 754–765.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of ACL-2014*, pages 271–281.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of EMNLP-2010*, pages 1223–1233.

Percy Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of ACL-2011*, pages 590–599.

Peter LoBue and Alexander Yates. 2011. Types of common-sense knowledge needed for Recognizing Textual Entailment. In *Proceedings of ACL/HLT-2011*, pages 329–334.

Bill MacCartney and Christopher Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of COLING-2008*, pages 521–528.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*, pages 140–156.

Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Department of Computer Science, Stanford University.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Ray Mooney. 2008. Learning to connect language and perception. In *Proceedings of AAAI-2008*, pages 1598–1601.

Lawrence Moss. 2010. Natural logic and semantics. In *Proceedings of 17th Amsterdam Colloquium*, pages 71–80.

Lenhart Schubert. 2015. Semantic representation. In *Proceedings of AAAI-2015*, pages 4132–4139.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP-2008*, pages 254–263.

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. *Proceedings of ACL-2014*, pages 79–89.

Johan van Benthem. 2008. A brief history of natural logic. Technical Report PP-2008-05, ILLC, University of Amsterdam.

Adrienne Wang, Tom Kwiatkowski, and Luke Zettlemoyer. 2014. Morpho-syntactic lexical generalization for CCG semantic parsing. In *Proceedings of EMNLP-2014*, pages 1284–1295.

Yuk Wah Wong and Raymond J. Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of HLT/NAACL-2006*, pages 439–446.

Luke S. Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of ACL-2009*, pages 976–984.