Dependency Parsing Features for Semantic Parsing

Will Monroe and **Yushi Wang**
Computer Science Department, Stanford University
Stanford, CA 94305
{wmonroe4,yushiw}@stanford.edu

Abstract

Current semantic parsing systems either largely ignore the syntactic structure of the natural language input or attempt to learn highly underconstrained, noisy syntactic representations. We present two new classes of features for statistical semantic parsing that utilize information about syntactic structure, extracted from a dependency parse of an utterance, to score semantic parses for that utterance. These features, when added to an existing state-of-the-art semantic parsing framework, improve macro-averaged F_1 from 31% to 38.3% over a strong baseline on a broad-coverage benchmark dataset. In experiments on small, focused datasets, we identify specific relationships between syntactic structure and semantic composition that these features enable the framework to learn, relationships it fails to learn without them.

1 Introduction

Semantic parsing is the task of converting natural language *utterances* into structured *meaning representations* (MRs). Construction of meaning representations is useful for a variety of applications, such as question answering (Clarke et al., 2010), dialogue systems (Artzi and Zettlemoyer, 2011), natural-language database interfaces (Zelle and Mooney, 1996), and natural-language instruction interpretation for robot control (Matuszek et al., 2012).

Most existing semantic parsing systems have not taken full advantage of readily available syntactic parsing systems. In our project we investigate using syntactic information in the form of the Stanford typed dependencies representation (de Marneffe and Manning, 2008) as part of the semantic parsing framework SEMPRE (Berant et al., 2013). We implement two new classes of features for scoring parse trees produced by SEMPRE that relate the structure of the parse tree and the rules used to construct it to the syntactic dependencies present in the utterance.

In order to illuminate the effects of these dependency parsing features, we construct two small, hand-built datasets and two medium-sized synthetic datasets that illustrate various linguistic properties that are captured by these features. We show that SEMPRE with our dependency parsing features added performs better on these datasets than the same system with a strong baseline feature set. We analyze the differences in performance on these datasets in detail, showing that that these features indeed capture useful relationships between syntactic and semantic structure that are not captured by the baseline feature set.

Finally, we perform experiments on the large, realistic, broad-coverage datasets FREE917 (Cai and Yates, 2013) and WEBQUESTIONS (Berant et al., 2013). The combination of our two classes of features improves macro-averaged F_1 on WEBQUESTIONS from 31% to 38.3% over the baseline, which includes features from the WEBQUESTIONS paper.

2 Related Work

For our work, we build on the SEMPRE system (Berant et al., 2013). SEMPRE learns to parse utterances into MRs from a training set of question-answer pairs, without requiring annotated logical forms. It treats the underlying logical forms as a latent variable and sums over them to define a probability distribution over answers. Given an input natural language sentence, SEMPRE considers all possible derivations (trees specifying a set of combination rules that culminate in some logical form for the sentence at the root) and selects the one with maximal likelihood. In training, it updates feature weights for all parses depending on whether they contribute to a derivation that gives the correct answer.

Like other state-of-the-art semantic parsing systems, SEMPRE uses a probabilistic grammar to de-

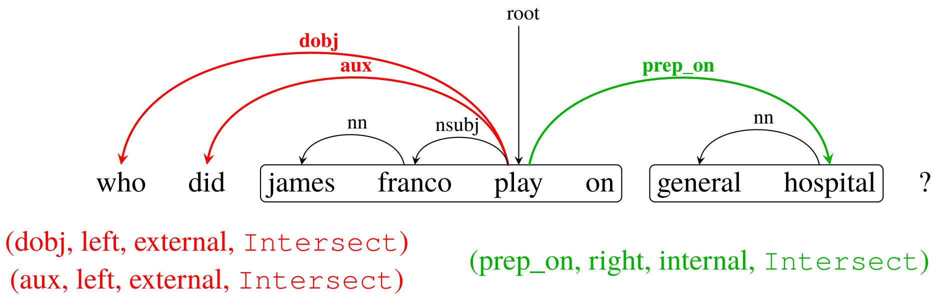
root
dobj
aux
prep_on
nn
nsubj
nn
who
did
james
franco
play
on
general
hospital
?
(dobj, left, external, Intersect)
(aux, left, external, Intersect)
(prep_on, right, internal, Intersect)

Figure 1: Composition dependency features, fired for a subderivation connecting the two circled spans with the composition function `Intersect`. Arcs that are used to create these features are highlighted.

fine a distribution over possible parses for the sentences and identify the most likely parse. Zettlemoyer and Collins (2005) learn a PCCG (Probabilistic Combinatory Categorical Grammar) from utterance–logical form pairs. Their system requires training set of these pairs in addition to a seed lexicon. The Scissor system (Ge and Mooney, 2005) maps sentences in natural language to MRs in the form of semantically augmented parse trees (SAPTs). A SAPT captures the semantic interpretation of individual words and the predicate-argument structure of the sentence. Wong and Mooney (2006) use SCFGs (synchronous context-free grammars), simultaneous rules for both the MR and natural language, in their system, the Word Alignment Semantic Parser (WASP). They create a lexicon of these rules to assemble MRs from sentences and apply a word alignment model to the problem of aligning words from the utterance to structural components in the MR.

While these systems learn relationships between the properties of the input utterance and properties of the correct parse trees and MRs, the properties considered for the utterance do not include useful prior knowledge about the structure of the natural language input that is expressed in a traditional syntactic parse. The grammars used by these systems are inspired by grammars of natural language. However, these grammars are often highly underconstrained; require large, noisy lexicons; and must be learned from much less data than is available for syntactic parsers. Furthermore, in each of the above approaches, the parsers all require a training set of sentences along with their annotated MRs.

3 Dependency-Based Features

We integrate features based on the Stanford typed dependencies representation (de Marneffe and Manning, 2008) into Sempre. Stanford Dependencies is a straightforward system for describing grammatical relations between heads and dependents in natural language sentences. The design choices made in creating Stanford Dependencies revolve around usability for broader CS community on tasks for which parsing is an intermediate step; here, our choice of Stanford Dependencies is motivated by their similarity to the meaning representations used in Sempre, which are an extension of Dependency-based Compositional Semantics (Liang et al., 2011).

In our extension to Sempre, we first annotate each utterance with a Stanford Dependencies tree, using the Stanford CoreNLP parser (Socher et al., 2013). Then, for each training example, we extract two types of indicator features that relate the dependency edges in the utterance to properties of candidate derivations produced for that utterance.

3.1 Composition features

The first class of features we extract relates the dependencies present in the sentence to the semantic grammar rules used in each subderivation of the correct parse.

Figure 1 shows the types of features included in this class. Specifically, given an input sentence and a candidate derivation, for each subderivation, we identify all dependencies that connect a word in one child of that subderivation to another child (internal dependencies) and all dependencies that connect some word within the span of the whole subderivation to a word in the rest of the sentence (external dependencies). We fire a feature for each such edge, specifying

- the dependency's grammatical relation,
- the direction (left or right) of the dependency,
- whether the dependency is internal or external, and

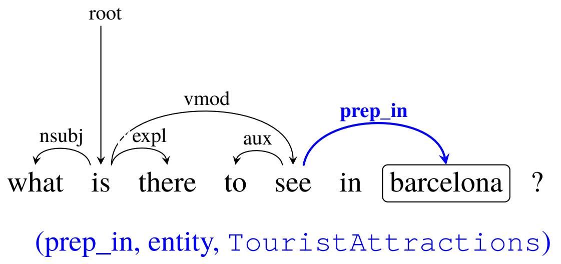
root
vmod
prep_in
nsubj
expl
aux
what is there to see in barcelona ?
(prep_in, entity, TouristAttractions)

Figure 2: A bridging dependency feature, fired for a subderivation bridging *barcelona* to the relation `TouristAttractions`. This is an example of the entity bridging function.

- the grammar rule being used to combine the spans.

These features capture information about grammatical relationships among the spans being combined and between the spans being combined and other words in the sentence.

3.2 Bridging features

We also implemented a set of features that fire only when a particular type of semantic composition function is used: the *bridging* functions, which create a relevant Freebase relation "out of thin air," in that it is not directly derived from lexical items in the utterance. This function addresses cases in which semantic relations are expressed by stop words (e.g., *[politicians] from [Maine]*) or not marked at all (e.g. *[Cher] [songs]*).

There are three types of bridging functions implemented in Sempre:

- *Entity* bridging inserts a relation extending outward from a single parsed span (usually a single Freebase entity). Example: *things to see in [Barcelona]*
- *Unary* bridging inserts a relation between two parsed spans, one of which is usually a single entity and the other of which is a more general set specified by a unary relation. Example: *[California] [national parks]*
- *Injection* inserts a modifier into a Freebase relation that takes more than two arguments (a *compound value type* or *mediator*). Example: *[{Fortune 500 rank} of {Google}] in [2010]*

Whenever one of these three bridging functions is used in a derivation, we construct a feature for each incoming dependency edge of the modifier phrase (the modifier phrases in the above examples are *Barcelona*, *California*, and *2010*, respectively) that includes

- the type of bridging (entity, unary, or inject),
- the dependency relation, and
- the Freebase relation inserted.

Figure 2 shows an example of a dependency feature extracted from an entity bridging operation.

4 Experiments

4.1 Baseline

We implement our features in the current development version of Sempre, which also serves as our baseline in evaluating the effectiveness of these features. We compare the performance of this development version to the performance of the same system with each of the above two classes of features, individually and combined.

The feature set used in this baseline system is based on the features used in Berant et al. (2013), which include:

- *Rule features*: indicator features over counts of grammar rules used in a derivation.
- *POS skipping features*: counts of words that are skipped in the derivation (those whose logical forms are ignored by some composition rule in the parse tree, and therefore don't participate in the final logical form), bucketed by part of speech.
- *POS joining features*: indicator features over the parts of speech of the first word in each of the two spans combined using the *join* operator, in conjunction with which predicate argument of the head is filled by the modifier.
- *Denotation features*: indicator features over the size of the denotation set produced by executing the logical form.

We refer the reader to Berant et al.'s paper for fuller descriptions of these features, as well as the definition of the join operator.

The baseline system also includes additional features that have been shown to be effective in further development of Sempre, such as Freebase popularity of relations and entities, string similarity between input tokens and Freebase relations, and the conjunction of denotation type and *wh-* question word introducing the utterance.

"Influence"	"Battle of New Orleans"
who was influenced by malcolm x?	*who fought in the battle of new orleans?*
who did malcolm x influence?	*where was the battle of new orleans fought?*
who was malcolm x influenced by?	*what was the site of the battle of new orleans?*
who was gandhi influenced by?	*who fought the battle of new orleans?*
who did gandhi influence?	*what site was the battle of new orleans fought in?*
who was influenced by gandhi?	*where was the battle of new orleans fought in?*

Table 1: Examples from the hand-crafted datasets. We include a sample of six of the 13 "influence" utterances and all six "Battle of New Orleans" utterances.

"Influence + TV"	"Play"
who influenced someone influenced by gandhi	*what sport does michael jordan play*
who was influenced by someone who malcolm x influenced	*who is played by leonard nimoy*
who influenced someone who was influenced by gandhi	*what do the houston rockets play*
who did the children of george bush influence	*what does wayne gretzky play*
who played kirk on star trek	*who is played by julia louis-dreyfus*
who did patrick stewart play on star trek	*who plays kirk on star trek*

Table 2: Six examples from each of the synthetic, PCFG-generated datasets. Many of the randomly-generated "influence + TV" examples were quite long; the utterances included here are chosen from among those that are short enough to fit on one line.

4.2 Datasets

In order to better understand the effect of these SD features, we ran SEMPRE with and without our dependency parsing features on six different datasets that illustrate different learning scenarios. First, we created two small, hand-constructed datasets that include examples with interesting grammatical dependencies:

- The **"influence"** dataset, which consists of several questions expressing the "influenced" and "influenced by" Freebase relations using various syntactic constructions.
- The **"Battle of New Orleans"** dataset, which consists of several wordings of two questions about said battle.

These datasets were preliminary experiments aimed at gauging the feasibility of using dependency features to improve upon SEMPRE's baseline. "Influence" tested the ability of the system to learn passive voice relations and differences in constituency structure. "Battle of New Orleans", with its long name that includes two internal dependency relations, was meant to test the ability of the system to parse compositional entity names.

We then used a probabilistic context-free grammar to randomly generate two medium-sized, semi-realistic datasets:

- The **"influence + TV"** dataset, which consists of longer "influence" sentences mixed in with questions about TV characters and actors.
- The **"play"** dataset, which contains multiple questions about various sports teams and players in addition to TV shows. Each of these utterances has some form of the verb "play" as its head.

These datasets test the capabilities of our new features in an expanded knowledge universe. The "influence + TV" dataset expands on the "influence" dataset, with much longer sentences and more compositional influence relationships, and has TV show questions mixed in to exercise the system's ability to generalize dependency relations across different verbs in the same lexicon. The "play" dataset tests the usefulness of dependency features for learning a single lexical item with multiple senses ("play" in the context of sports and TV show roles) and multiple arguments expressed by different dependency relations (nsubj/agent: actor; dobj/nsubjpass: character; prep_on: show).

Examples of utterances from each of our datasets are shown in Tables 1 and 2. For the "Battle of New Orleans" dataset, we include the entirety of the six-utterance dataset; the other tables show a sample of six utterances from each dataset.

Finally, we ran experiments on two large, realistic, general-purpose datasets: FREE917 (Cai and Yates, 2013) and WEBQUESTIONS (Berant et al., 2013). Both of these large datasets were used in Berant et al.'s evaluation of SEMPRE, and offer a way to compare SEMPRE's performances with and without our modifications.

	I	LE	T+I	P
Size	7	3	14	24
Baseline	2	1	6	21
OrderedPOS	**7**	2	10	15
+ D/C	**7**	**3**	***12***	***22***
+ D/B	**7**	**3**	10	21
+ D/C, D/B	**7**	**3**	11	***22***

Table 3: Number of utterances parsed to a correct denotation by SEMPRE with different features on various hand-constructed or synthetic datasets. **D/C** is the composition dependency features; **D/B** is the bridging dependency features. Small datasets: **I** (influence) and **LE** (long entity); synthetic datasets: **T+I** (TV + influence) and **P** ("play"); large datasets: **F917** (FREE917) and **WQ** (WEBQUESTIONS).

	FREE917	WEBQUESTIONS
Size	129	756
Baseline	71.3	32.9
OrderedPOS	***72.9***	31.0
+ D/C	66.7	36.1
+ D/B	66.7	30.3
+ D/C, D/B	69.0	***38.3***

Table 4: Performance on large datasets (development set). Following Berant et al. (2013), we report accuracy (%) on FREE917 and macro-averaged F_1 (%) on WEBQUESTIONS.

4.3 Evaluation metrics

For our experiments on hand-constructed and synthetic data, we report denotation accuracy (percentage of questions for which the system returns exactly the correct denotation). For the larger datasets, we use the same evaluation metrics used in Berant et al. (2013): denotation accuracy for FREE917, and macro-averaged F_1 for WEBQUESTIONS. The F_1 metric is more appropriate for WEBQUESTIONS because the gold denotations are produced by Mechanical Turk annotators and are not guaranteed to be the result of executing a specific logical form, unlike in FREE917.

5 Results and Analysis

5.1 Hand-constructed data

We found that SEMPRE with the composition dependency features in Section 3.1 was able to score 100% on the "influence" dataset using an associated toy grammar, but without the new features it was unable to correctly answer several examples. In analyzing the errors SEMPRE made on this dataset with its original feature set, however, we found that an existing feature could be modified to fix all of the errors without using dependency parsing. This feature related the parts of speech of the first word in each span to the argument structure of the logical form being assembled. When modified to include the *order* in which the two POS-tagged words occurred, this feature was able to yield 100% accuracy on this toy dataset without dependency features.

In the "Battle of New Orleans" dataset, SEMPRE without dependency parsing features fails to generalize to one of the examples on the test set that it is able to get correct with them. This example is *where was the battle of new orleans fought?* The answer it gives is the combatants rather than the location. From the training data, the system without dependency parsing learns to associate the when *fought* when not followed by *in* with the combatants relation.

With dependency parsing, the system shifts weight to features that fire in the presence of the *det* edge from *Battle* to the definite article and a following binary relation, influencing the system to avoid assigning the noun phrase to the second argument of the relation. The fact that this was necessary to arrive at a correct parse revealed a fact about the hard-coded lexicon that we used for this dataset, namely that the two similar relations ("combatants" and "event location") had the opposite place structure for their two arguments. Though this could be seen as a bug in our lexicon, such inconsistencies are ubiquitous in broad-coverage lexicons, and in this case the system with dependency features was able to handle this messiness in a way that generalized to unseen data, albeit by using a very shallow property of the training data.

5.2 Synthetic data

In the "TV + influence" dataset, SEMPRE is unable to learn to correctly parse a particular class of utterances without dependency features, those of the form *who played (character) on (show)*. This is due to the fact that there are more influence utterances in the dataset than TV utterances. The abundance of the influence utterances and their relation argument order caused the learner to weight the ordered POS feature in such a way that it penalizes assigning a following noun phrase to the object of a past tense verb. Including dependency parsing features allowed the system to make the richer syntactic distinctions (beyond word order and parts of speech) that are necessary to correctly parse a few of the examples in the test set.

On the "play" dataset, we focus on a single verb to emphasize the capability of the dependency parsing features to learn how to assign semantic roles to various constituents of that verb, for two different senses of the verb. The class of utterances missed without dependency features on this dataset are those of the form *who is played by (actor)*; for these, the baseline feature set learns that the answer should be one entity, but for these questions, the answer is usually more than one character. The dependency parsing features, as intended, are able to associate dependency relations with the various arguments of Freebase relation.

This result suggests an avenue of future work in the format of the lexicon. Currently, lexical items are stored as a list of phrase–logical form pairs. This could be enhanced by replacing the role of contiguous phrases with that of lexical dependency paths (for example, $\left[\mathit{play} \xrightarrow{\text{prep\_on}}\right]$ instead of [play on]).

5.3 Larger datasets

On the Free917 dataset, the improved POS-linking feature resulted in a small improvement (72.9% compared to 71.3% with a baseline feature set). We found that including the dependency features increased training set performance slightly on Free917 (81.1% vs. 76.6%) but was detrimental to development set performance (69.0% vs. 72.9%), suggesting that adding in our new features resulted in overfitting the data.

Among the examples that our dependency features improved performance on, many involved long entities with one or more common content words:

- what's the focus of the *last minute blog*
- in which *comic book issue* did kitty pryde first appear

Similar to the intent of the "Battle of New Orleans" dataset, these longer entities contain internal dependencies, allowing for our dependency parser to better recognize and correctly parse these entities, whereas parsing without these features may result in misinterpreting the entities' spans. For the most part, however, adding dependency features resulted in overfitting the training data. Several examples that had been parsed correctly by the baseline features were parsed incorrectly, due to the noise from the extra features.

For example, in all runs with composition dependency features, all questions about cause of death (*where did nathan smith die*, *how did samuel beckett die*) were parsed incorrectly. Inspecting the data reveals that in the training data, there were no questions about cause of death. This resulted in several negative weights being associated with features relating common dependency structure patterns to the "cause of death" relation.

On WebQuestions, we found that the composition dependency features improved performance over the baseline feature set, increasing macro-averaged F_1 from 31% to 36.1%. Including only the bridging features did not yield any improvement (30.3%), but including both classes of features further improved F_1, to 38.3%.

We found that the set of examples on which the system improved after including composition dependency features included a large number of utterances ending in a preposition. Examples include:

- *what city was leonardo da vinci from?*
- *what county is sacramento located in?*
- *what did lucille ball die of?*
- *what school did sir isaac newton go to?*
- *what show is jill wagner on?*

On the other hand, the baseline system tended to fare better or equally well on questions that are comparatively lacking in information from dependency relations. These questions included many of the format "what is __?", such as:

- *what is the political system in egypt?*
- *what is pennsylvania's state flower called?*

In these questions, the additional features provided little to no benefit, and often lowered accuracy due to the additional noise it provided, sometimes causing correct parses to be discarded in the beam search.

6 Conclusion

In this work we examine ways in which semantic composition relates to grammatical dependency relations. We show that features utilizing information present in the dependency structure of utterances helps Sempre learn various aspects of the relationship between syntax and logical form. In the process, we also find one simple improvement that can make an existing feature in SEMPRE more powerful without requiring the addition of a dependency parser. This improved feature uses fine-grained parts of speech and constituent order to enable the system to learn whether two phrases should be linked using a semantic composition rule.

We identify three particular scenarios in which dependency-based features enable SEMPRE to learn semantic-syntactic correlations present in a dataset that it could not otherwise learn. First, we show that in the face of a messy or incomplete lexicon in a low-data setting, dependency-based features help SEMPRE identify shallow properties of the input utterance that generalize well. Second, we demonstrate that dependency-based features allow the system to learn a mapping from verb-dependency pairs to Freebase logical predicates. Third, we present empirical results that suggest that dependency parsing features help with identifying the boundaries of multi-word entities and choosing correct Freebase relations for utterances with sentence-final prepositions.

We have contributed our features to the SEMPRE codebase for future work in semantic parsing.

Contributions

Both team members contributed equally to all aspects of the project.

Acknowledgments

We thank Percy Liang for close advising throughout this project, and Jonathan Berant for helpful correspondence in working with SEMPRE. We also thank the course instructors for the excellent, inspiring course.

References

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *EMNLP*.

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

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL*.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *CoNLL*.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *CoLING 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*.

Ruifang Ge and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *CoNLL*.

Percy Liang, Michael I. Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *ACL*.

Cynthia Matuszek, Nicholas Fitzgerald, Luke Zettlemoyer, Liefeng Bo, and Dieter Fox. 2012. A joint model of language and perception for grounded attribute learning. In *ICML*.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with compositional vector grammars. In *ACL*.

Yuk Wah Wong and Raymond J. Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *NAACL-HLT*.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI*, pages 1050–1055.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*.