

# An Algorithm for Open Text Semantic Parsing

Lei Shi and Rada Mihalcea  
Department of Computer Science  
University of North Texas  
leishi@unt.edu, rada@cs.unt.edu

## Abstract

This paper describes an algorithm for open text shallow semantic parsing. The algorithm relies on a frame dataset (FrameNet) and a semantic network (WordNet), to identify semantic relations between words in open text, as well as shallow semantic features associated with concepts in the text. Parsing semantic structures allows semantic units and constituents to be accessed and processed in a more meaningful way than syntactic parsing, moving the automation of understanding natural language text to a higher level.

## 1 Introduction

The goal of the semantic parser is to analyze the semantic structure of a natural language sentence. Similar in spirit with the syntactic parser – whose goal is to parse a valid natural language sentence into a parse tree indicating how the sentence can be syntactically decomposed into smaller syntactic constituents – the purpose of the semantic parser is to analyze the structure of sentence meaning. Sentence meaning is composed by *entities* and *interactions* between entities, where entities are assigned semantic *roles*, and can be further modified by other *modifiers*. The meaning of a sentence is decomposed into smaller semantic units connected by various semantic relations by the principle of compositionality, and the parser represents the semantic structure – including semantic units as well as semantic relations, connecting them into a formal format.

In this paper, we describe the main components of the semantic parser, and illustrate the basic procedures involved in parsing semantically open text. We believe that such structures, reflecting various levels of semantic interpretation of the text, can be used to improve the quality of text processing applications, by taking into account the meaning of text.

The paper is organized as follows. We first describe the semantic structure of English sentences, as the basis for semantic parsing. We then intro-

duce the knowledge bases utilized by the parser, and show how we use this knowledge in the process of semantic parsing. Next, we describe the parsing algorithm and elaborate on each of the three main steps involved in the process of semantic parsing: (1) syntactic and shallow semantic analysis, (2) semantic role assignment, and (3) application of default rules. Finally, we illustrate the parsing process with several examples, and show how the semantic parsing algorithm can be integrated into other language processing systems.

## 2 Semantic Structure

Semantics is the denotation of a string of symbols, either a sentence or a word. Similar to a syntactic parser, which shows how a larger string is formed by smaller strings from a formal point of view, the semantic parser shows how the denotation of a larger string – sentence, is formed by denotations of smaller strings – words. Syntactic relations can be described using a set of rules about how a sentence string is formally generated using word strings. Instead, semantic relations between semantic constituents depend on our understanding of the world, which is across languages and syntax.

We can model the sentence semantics as describing entities and interactions between entities. Entities can represent physical objects, as well as time, places, or ideas, and are usually formally realized as nouns or noun phrases. Interactions, usually realized as verbs, describe relationships or interactions between participating entities. Note that a participant can also be an interaction, which can be regarded as an entity nominalized from an interaction. We assign semantic roles to participants, and their semantic relations are identified by the case frame introduced by their interaction. In a sentence, participants and interactions can be further modified by various modifiers, including descriptive modifiers that describe attributes such as *drive slowly*, restrictive modifiers that enforce a general denotation to become more specific such as *musical instrument*, referential modifiers that indicate partic-

ular instances such as *the pizza I ordered*. Other semantic relations can also be identified, such as coreference, complement, and others. Based on the principle of compositionality, the sentence semantic structure is recursive, similar to a tree.

The semantic parser analyzes shallow-level semantics, which is derived directly from linguistic knowledge, such as rules about semantic role assignment, lexical semantic knowledge, and syntactic-semantic mappings, without taking into account any context or common sense knowledge. The parser can be used as an intermediate semantic processing tool before higher levels of text understanding.

### 3 Knowledge Bases for Semantic Parsing

One major problem faced by many natural language understanding applications that rely on syntactic analysis of text, is the fact that similar syntactic patterns may introduce different semantic interpretations. Likewise, similar meanings can be syntactically realized in many different ways. The semantic parser attempts to solve this problem, and produces a syntax-independent representation of sentence meaning, so that semantic constituents can be accessed and processed in a more meaningful and flexible way, avoiding the sometimes rigid interpretations produced by a syntactic analyzer. For instance, the sentences *I boil water* and *water boils* contain a similar relation between *water* and *boil*, even though they have different syntactic structures.

To deal with the large number of cases where the same syntactic relation introduces different semantic relations, we need knowledge about how to map syntax to semantics. To this end, we use two main types of knowledge – about words, and about relations between words. The first type of knowledge is drawn from WordNet – a large lexical database with rich information about words and concepts. We refer to this as *word-level knowledge*. The latter is derived from FrameNet – a resource that contains information about different situations, called *frames*, in which semantic relations are syntactically realized in natural language sentences. We call this *sentence-level knowledge*. In addition to these two lexical knowledge bases, the parser also utilizes a set of manually defined rules, which encode mappings from syntactic structures to semantic relations, and which are also used to handle those structures not explicitly addressed by FrameNet or WordNet.

In this section, we describe the type of information extracted from these knowledge bases, and show how this information is encoded in a format

accessible to the semantic parser.

#### 3.1 Frame Identification and Semantic Role Assignment

FrameNet (Johnson et al., 2002) provides the knowledge needed to identify case frames and semantic roles. FrameNet is based on the theory of frame semantics, and defines a sentence level ontology. In frame semantics, a *frame* corresponds to an *interaction* and its *participants*, both of which denote a scenario, in which participants play some kind of roles. A frame has a name, and we use this name to identify the semantic relation that groups together the semantic roles. In FrameNet, nouns, verbs and adjectives can be used to identify frames.

Each annotated sentence in FrameNet exemplifies a possible syntactic realization for the semantic roles associated with a frame for a given target word. By extracting the syntactic features and corresponding semantic roles from all annotated sentences in the FrameNet corpus, we are able to automatically build a large set of rules that encode the possible syntactic realizations of semantic frames.

In our implementation, we use only verbs as target words for frame identification. Currently, FrameNet defines about 1700 verbs attached to 230 different frames. To extend the parser coverage to a larger subset of English verbs, we are using VerbNet (Kipper et al., 2000), which allows us to handle a significantly larger set of English verbs.

VerbNet is a verb lexicon compatible with WordNet, but with explicitly stated syntactic and semantic information using Levin’s verb classification (Levin, 1993). The fundamental assumption is that the syntactic frames of a verb as an argument-taking element are a direct reflection of the underlying semantics. Therefore verbs in the same VerbNet class usually share common FrameNet frames, and have the same syntactic behavior. Hence, rules extracted from FrameNet for a given verb can be easily extended to verbs in the same VerbNet class. To ensure a correct outcome, we have manually validated the FrameNet-VerbNet mapping, and corrected the few discrepancies that were observed between VerbNet classes and FrameNet frames.

##### 3.1.1 Rules Learned from FrameNet

FrameNet data “is meant to be lexicographically relevant, not statistically representative” (Johnson et al., 2002), and therefore we are using FrameNet as a starting point to derive rules for a rule-based semantic parser.

To build the rules, we are extracting several syntactic features. Some are explicitly encoded in FrameNet, such as the *grammatical function* (GF)

and *phrase type* (PT) features.

In addition, other syntactic features are extracted from the sentence context. One such feature is the *relative position* (RP) to the target word. Sometimes the same syntactic constituent may play different semantic roles according to its position with respect to the target word. For instance the sentences: *I pay you.* and *You pay me.* have different roles assigned to the same lexical unit *you* based on the relative position with respect to the target word *pay*.

Another feature is the *voice* of the sentence. Consider these examples: *I paid Mary 500 dollars.* and *I was paid by Mary 500 dollars.* In these two sentences, *I* has the same values for the features GF, PT and RP, but it plays completely different roles in the same frame because of the difference of voice.

If the phrase type is prepositional phrase (PP), we also record the actual *preposition* that precedes the phrase. Consider these examples: *I was paid for my work.* and *I was paid by Mary.* The prepositional phrases in these examples have the same values for the features GF, PT, and RP, but different prepositions differentiate the roles they should play.

After we extract all these syntactic features, the semantic role is appended to the rule, which creates a mapping from syntactic features to semantic roles.

Feature sets are arranged in a list, the order of which is identical to that in the sentence. The order of sets within the list is important, as illustrated by the following example: “*I give the boy a ball.*” Here, *the boy* and *a ball* have the same features as described above, but since *the boy* occurs before *a ball*, then *the boy* plays the role of *recipient*. Altogether, the rule for a possible realization of a frame exemplified by a tagged sentence is an ordered sequence of syntactic features with their semantic roles.

For instance, Table 1 lists the syntactic and semantic features extracted from FrameNet for the sentence *I had chased Selden over the moor.*

	I	had chased	Selden	over the moor
GF	Ext	obj	comp	
PT	NP	Target	NP	PP
Position	before		after	after
Voice		active		
PP				over
Role	Theme		Goal	Path

Table 1: Example sentence with syntactic and semantic features

The corresponding formalized rule for this sentence is:

```
[active, [ext,np,before,theme], [obj,np,
after,goal], [comp,pp,after,over,path]]
```

In FrameNet, there are multiple annotated sentences for each frame to demonstrate multiple possible syntactic realizations. All possible realizations of a frame are collected and stored in a list for that frame, which also includes the target word, its syntactic category, and the name of the frame. All the frames defined in FrameNet are transformed into this format, so that they can be easily handled by the rule-based semantic parser.

### 3.2 Word Level Knowledge

WordNet (Miller, 1995) is the resource used to identify shallow semantic features that can be attached to lexical units. For instance, attribute relations, adjective/adverb classifications, and others, are semantic features extracted from WordNet and stored together with the words, so that they can be directly used in the parsing process.

All words are uniformly defined, regardless of their class. Features are assigned to each word, including syntactic and shallow semantic features, indicating the functions played by the word. Syntactic features are used by the feature-augmented syntactic analyzer to identify grammatical errors and produce syntactic information for semantic role assignment. Semantic features encode lexical semantic information extracted from WordNet that is used to determine semantic relations between words in various situations.

Features can be arbitrarily defined, as long as there are rules to handle them. The features we define encode information about the syntactic category of a word, number and countability for nouns, transitivity and form for verbs, type, degree, and attribute for adjectives and adverbs, and others. Table 2 lists the main features used for content words.

Feature	Values
Nouns	
Number	singular/plural
Countability	countable/uncountable
Verbs	
Transitivity	transitive/intransitive/double transitive
Form	normal/infinitive/present participle/past participle
Adjectives	
Type	descriptive/restrictive/referential
Attribute	arbitrary
Degree	base/comparative/superlative
Adverbs	
Type	descriptive/restrictive/referential
Attribute	arbitrary
Degree	base/comparative/superlative

Table 2: Features for content words

For example, for the word *dog*, the entry in the

lexicon is defined as:

```
lex(dog,W):- W= [parse:dog, cat:noun,
  num:singular, count:countable].
```

Here, the category (*cat*) is defined as noun, the number (*num*) is singular, and we also record the countability (*count*)<sup>1</sup>.

For adjectives, the value of the *attribute* feature is also stored, which is provided by the *attribute* relation in WordNet. This relation links a descriptive adjective to the attribute (noun) it modifies, such as *slow* → *speed*. For example, for the adjective *slow*, the entry in the lexicon is defined as:

```
lex(slow,W):- W= [parse:slow, cat:adj,
  attr:speed, degree:base, type:descriptive].
```

Here, the category (*cat*) is defined as adjective, the *type* is descriptive, *degree* is base form. We also record the *attr* feature, which is derived from the *attribute* relation in WordNet, and links a descriptive adjective to the attribute (noun) it modifies, such as *slow* → *speed*.

We are also exploiting the transitional relations from adverbs to adjectives and to nouns. We noticed that some descriptive adverbs have correspondence to descriptive adjectives, which in turn are linked to nouns by the attribute relation. Using these transitional links, we derive relations like: *slowly* → *slow* → *speed*. A typical descriptive adverb is defined as follows:

```
lex(slowly,W):- W= [parse:slowly, cat:adv,
  attr:speed, degree:base, type:descriptive].
```

In addition to incorporating semantic information from WordNet into the lexicon, this word level ontology is also used to derive default rules, as discussed later.

### 3.3 Hand-coded Knowledge

The FrameNet database encodes various syntactic realizations only for semantic roles within a frame. Syntax-semantics mappings other than semantic roles are manually encoded as rules integrated in the syntactic-semantic analyzer. The analyzer determines the syntactic structure of the sentence, and once a particular syntactic constituent is identified, its corresponding mapping rules are immediately applied. The syntactic constituent is

---

<sup>1</sup>The value for the feature (countability) is obtained from word properties stored in the Link parser dictionaries (<http://www.link.cs.cmu.edu/link/>). The Link dictionaries are also used to derive the lists of words to be stored in the lexicon. Note however that the Link parser itself is not used in the parsing process.

then translated into its corresponding semantic constituent, together with the relevant semantic information.

Some semantic relations can be directly derived from syntactic patterns. For example, a restrictive relative clause such as “the man *that you see*” serves as a referential modifier. An adverbial clause beginning with “because” is a modifier describing the “reason” of the interaction. The inflection from “apple” to “apples” adds an attributive modifier of *quantity* to the entity “apple”.

However, syntactic relations may often introduce semantic ambiguity, with multiple possible interpretations. To handle these cases, we encode rules that describe all possible interpretations of any given structure, and then use lexical semantic information as selectional restrictions for ambiguity resolution. For instance, in “a book on Chinese history”, *on Chinese history* describes the topic of the book and this interpretation can be uniquely determined by noting that *history* is not a physical object, and thus the interpretation of *on Chinese history* as describing location is semantically anomalous. Instead, in “a book on the computer”, *on the computer* may describe a location, but it could also describe the book topic, and hence the correct interpretation of this sentence cannot be determined without additional context. In such cases, the semantic parser produces all possible interpretations, allowing systems that use the semantic parser’s output to determine the right interpretation that best fits the application at hand.

Selectional restrictions – as part of the hand-coded knowledge – are used for both semantic role identification and syntax-semantics translation. These additional rules are needed to supplement the information encoded in FrameNet, since FrameNet only annotates syntactic features, which often times do not provide enough information for identifying correct semantic roles.

Consider for example “I break the window” vs. “The hammer breaks the window”. According to our semantic parser, the participants in the interaction “break” have exactly the same syntactic features in both sentences, but they play different semantic roles (“I” plays the *agent* role while “hammer” plays the *instrument* role), since they belong to different ontological categories: “I” refers to a person and “hammer” refers to a tool. This interpretation is not possible using only FrameNet information, and thus we fill the gap by attaching selectional restrictions to the rules extracted from FrameNet.

The definition of selectional restriction is based on WordNet 2.0 noun hierarchy. We say that en-

tivity  $E$  belongs to the ontological category  $C$  if the noun  $E$  is a child node of  $C$  in the WordNet semantic hierarchy of nouns. For example, if we define the ontological category for the role “instrument” as *instrumentality*, then all hyponyms of *instrumentality* can play this role, while other nouns like “boy”, which are not part of the *instrumentality* category will be rejected. Selectional restrictions are defined using a Disjunctive Normal Form (DNF) in the following format:

```
[Onto(ID,P),Onto(ID,P),...],[Onto(ID,P),...],...
```

Here, “Onto” is a noun and ID is its WordNet sense, which uniquely identifies Onto as a node in the semantic network. “P” can be set to  $p$  (positive) or  $n$  (negative), denoting if a noun should belong to the given category or not. For example, [person(1,n),object(1,p)],[substance(1,p)] means that the noun should belong to object(sense #1) but not person(sense #1)<sup>2</sup>, or it should belong to substance(sense #1). This information is added to the rules derived from FrameNet, and therefore after this step, a complete FrameNet rule entry is:

```
[Voice,[GF,PT,SelectionalRestriction,Role],...].
```

## 4 Semantic Parsing

The general procedure of semantic parsing consists of three main steps<sup>3</sup>: (1) The syntactic-semantic analyzer analyzes the syntactic structure, and uses hand-coded rules as well as lexical semantic knowledge to identify some semantic relations between constituents. It also prepares syntactic features for semantic role assignment in the next step. (2) The role assigner uses rules extracted from FrameNet, and assigns semantic roles for identified participants, based on their syntactic features as produced in the first step. (3) For those constituents not exemplified in FrameNet, we apply default rules to decide their default meaning.

### 4.1 Feature Augmented Syntactic-Semantic Analyzer

The analyzer is implemented as a bottom-up chart parsing algorithm based on features. We include rules of syntax-semantics mappings in the unification based formalism. The parser analyzes syntactic relations and immediately applies corresponding mapping rules to obtain semantic relations when a

<sup>2</sup>person(sense #1) is a child node of *object*(sense #1) in WordNet

<sup>3</sup>The parsing algorithm is implemented as a rule-based system using a declarative programming language Prolog.

syntactic relation is identified. Most semantic relations (e.g. various modifiers) are identified in this step, except semantic role annotation and application of default rules, which are postponed for a later stage. The analyzer generates an intermediate format, where target words and arguments are explicitly tagged with their syntactic and semantic features, so that they can be matched against the rules derived from FrameNet. We are using a feature-based analyzer that accomplishes three main tasks:

#### 4.1.1 Check if the sentence is grammatically correct

The syntactic analyzer is based on a feature augmented grammar, and therefore has the capability of detecting if a sentence is grammatically correct (unlike statistical parsers, which attempt to parse any sentence, regardless of their well-formedness). The grammar consists of a set of rules defining how constituents with different syntactic or semantic features can unify with each other.

By defining a grammar in this way, using features, once the right features are selected, the analyzer can reject some grammatically incorrect sentences such as: *I have much apples.*, *You has my car.*, and some semantically anomalous sentences: *The technology is very military.*<sup>4</sup>.

#### 4.1.2 Provide features for semantic role assignment

Through syntactic-semantic analysis in the first step, sentences are transformed into a format in which target words and syntactic constituents are explicitly tagged with their features. Unlike FrameNet – which may also assign roles to adverbs, we only use the subject, object(s) and prepositional phrases as potential participants in the interaction for semantic role labeling<sup>5</sup>. The analyzer marks verbs as *target* words for frame identification, identifies constituents for semantic role assignment, and produces features such as GF, PT, Voice, Preposition, as well as ontological categories for each constituent, in a format identical to the rules extracted from FrameNet, so that they can be matched with the frame definitions.

The ontological categories of constituents are used to match selectional restrictions, and are automatically derived from the head word of the noun phrase, or the head word of the noun phrase of the prepositional phrase. For other constituents that act like nouns, such as pronouns, infinitive forms, gerunds, or noun clauses, we have manually defined

<sup>4</sup>Since *military* is not a descriptive adjective, it cannot be modified by *very* and predicative use is forbidden.

<sup>5</sup>Adverbs are treated as modifiers.

ontological categories. For example, “book” is the ontological category of the phrase “the interesting book” and “on the book”. “person” is the ontological category we manually define for the pronoun “he”. We have also defined several special ontological categories that are not in WordNet such as *any*, which can be matched to any selectional restriction, *nonperson*, which means everything except *person*, and others. Note that this matching procedure also plays the role of a word sense disambiguation tool, by selecting only those categories that match the current frame constituents. After this step, target words and syntactic constituents can be assigned with the corresponding case frame and semantic roles during the second step of semantic parsing.

### 4.1.3 Identify some semantic relations

Some semantic relations can be identified in this phase. These semantic relations include word level semantic relations, and some semantic relations that have direct syntactic correspondence by using syntax-semantics mapping rules. This phase can also identify the function of the sentence such as *assertion*, *query*, *yn-query*, *command* etc, based on syntactic patterns of the sentence.

The output of the analyzer is an intermediate format suitable for the semantic parser, which contains syntactic features and identified semantic relations. For example, the output for the sentence “*He kicked the old dog.*” is:

```
[assertion,
 [[tag, ext, np, person,
  [[entity, [he], reference(third)],
   [modification(attribute), quantity(single)],
   [modification(attribute), gender (male)]],
 [target, v, kick, active, [kick]],
  [modification(attribute), time (past)],
 [tag, obj, np, dog,
  [[modification(reference), reference(the)],
   [modification(attribute), age(old)],
 [target, n, dog, [dog]]]]]]
]
```

## 4.2 Semantic Role Assignment

In the process of semantic role assignment, we first start by identifying all possible frames, according to the *target* word. Next, a matching algorithm is used to find the most likely match among all rules of these frames, to identify the correct frame (or frames if several are possible), and assign semantic roles.

In a sentence describing an interaction, we select the verb as the *target* word, which triggers the sentence level frame and uses the FrameNet rules of that target word for matching. If the verb is not

defined in FrameNet and VerbNet, we use WordNet synonymy relation to check if any of its synonyms is defined in FrameNet or VerbNet. If such synonyms exist, their rules are applied to the target word. This approach is based on the idea introduced by Levin that “what enables a speaker to determine the behavior of a verb is its meaning” (Levin, 1993). Synonymous verbs always introduce the same semantic frame and usually have the same syntactic behavior. To minimize information in the verb lexicon, non-frequently used verbs usually inherit a subset of the syntactic behavior of their frequently used synonyms. Since VerbNet has defined a framework of syntactic-semantic behavior for these frequently used verbs, the behavior of other related verbs can be quite accurately predicted by using WordNet synonymy relations. Using this approach, we achieve a coverage of more than 3000 verbal lexical units.

The matching algorithm relies on a scoring scheme to evaluate the similarity between two sequences of features. The matching starts from the first constituent of the sentence. It looks through the list of entries in the rule and when a match is found, it moves to the next constituent looking for a new match. A match involves match of syntactic features, as well as match of selectional restrictions. An exact match means that both syntactic features and selectional restrictions are matched, which increments the score of matching by 3. We apply selectional restriction by looking up the WordNet noun hierarchies. If the node of the ontological category is within the areas that the selectional restriction describes, this is regarded as a match. When applying selectional restrictions, due to polysemy of the ontological entries, we try all possible senses, starting from the most frequently used sense according to WordNet, until one sense meets the selectional restriction. If the syntactic features match exactly, but none of the possible word senses meet the selectional restrictions, this is regarded as a partial match, which increments the score by 2.

Partial matching is also possible, for a relaxed application of selectional restriction. This enables anaphora and metaphor resolution, in which the constituents have either unknown ontological category, or inherit features from other ontological categories (by applying high level knowledge such as personification). The number of subjects and objects as well as their relative positions should be strictly obeyed, since any variations may result in significant differences for semantic role labeling. Prepositional phrases are free in their location because the preposition is already a unique identi-

fier. Finally, after all constituents have found their match, if there are still remaining entries in the rule, the total score is decreased by 1. This is a penalty paid by partial matches, since additional constituents may indicate different semantic role labeling, which may change the interpretation of the entire sentence.

A polysemous verb may belong to multiple frames, and a frame pertaining to a given target word may have multiple possible syntactic realizations, exemplified by different sentences in the corpus. We try to match the syntactic features in the intermediate format with all the rules of all the frames available for the target word, and compare their matching scores. The rule with the highest score is selected, and used for semantic role assignment. Through this scoring scheme, the matching algorithm tries to maximize the utilization of syntactic and semantic information available in the sentence, to correctly identify case frames and semantic roles.

#### 4.2.1 Walk-Through Example

Assume the following two rules, triggered for the target word *break*:

```
1: [active, [ext, np, [[person(1,p)]]], agent],
   [obj, np, [[object(1,p)]]], theme],
   [comp, pp, with, [[instrumentality(3,p)]]],
   instrument]]
2: [[ext, np, [[instrumentality(3,p)]]], instrument],
   [obj, np, [[person(1,n), object(1,p)]]], theme]]
3: [[ext, np, [[person(1,n), object(1,p)]]], theme]]
```

And the sentences:

A: *I break the window with a hammer*

B: *The hammer breaks the window*

C: *The window breaks on the wall*

The features identified by the analyzer are:

```
A': [[ext, np, active, person],
     [obj, np, active, window],
     [comp, pp, active, with, hammer]]
B': [[ext, np, active, hammer],
     [obj, np, active, window]]
C': [[ext, np, active, window],
     [comp, pp, on, wall]]
```

Using the matching/scoring algorithm, the score for matching A' to rule 1 is determined as 9 since there are 3 exact matches, and to rule 2 as 5 since there is an exact match for “the window” but a partial match for “I”. Hence, the matching algorithm selects rule 1, and the semantic role for “I” is *agent*. Similarly, when we match B' to rule 1, we obtain a score of 4, since there is an exact match for “the window”, a partial match for “the hammer”, and

rule 1 has an additional entry for a prepositional phrase, which decrements the score by 1. It makes a larger score of 6 for matching with rule 2. Therefore, for the second case, the role assigned to “the hammer” is *instrument*. Rule 3 is not applied to the first two sentences since they have additional objects; similarly, rule 1 and 2 cannot be applied to sentence C for the same reason. The first constituent in C finds an exact match in rule 3 with a total score of 3, and hence “the window” is assigned the correct role *theme*. The prepositional phrase “on the wall”, for which no entry for labeling a role is found in rule 3, will be handled by default rules (see Section 4.3).

Based on the principle of compositionality, modifiers and constituents assigned semantic roles can describe interactions, so the semantic role assignment is performed recursively, until all roles within frames triggered by all target words are assigned.

#### 4.3 Applying Default Rules

We always assign semantic roles to subjects and objects<sup>6</sup>, but only some prepositional phrases can introduce semantic roles, as defined in the FrameNet case frames. Other prepositional phrases function as modifiers; in order to handle these constituents, and allow for a complete semantic interpretation of the sentence, we have defined a set of default rules that are applied as the last step of the semantic parsing process. For example, FrameNet defines a role for the prepositional phrase *on him* in “I depend *on him*” but not for *on the street* in “I walk *on the street*”, because it does not play a role, but it is a modifier describing a location. Since the role for the prepositional phrase beginning with *on* is not defined for the target word *walk* in FrameNet, we apply the default rule that “on something” modifies the *location* attribute of the interaction *walk*. Note that we include selectional restriction in the default rule since constituents with the same syntactic features such as “on Tuesday” and “on the table” may have obviously different semantic interpretations. An example of a default rule is shown below, indicating that the interpretation of a prepositional phrase followed by a time period (where *time\_period* is an ontological category from WordNet) is that of time modifier:

```
DefaultRule([_, pp, _, on, Onto, _], time):-
  SelectionalRestriction
    (Onto, 1,
     [[time_period(1,p)]])
```

We have defined around 100 such default rules, which are applied during the last step of the seman-

<sup>6</sup>Where a subject and object are usually realized by noun phrases, noun clauses, or infinitive forms.

tic parsing process.

## 5 Parser Output and Evaluation

We illustrate here the output of the semantic parser on a natural language sentence, and show the corresponding semantic structure and tree<sup>7</sup>. For example, for the sentence *I like to eat Mexican food because it is spicy*, the semantic parser produces the following encoding of sentence type, frames, semantic constituents and roles, and various attributes and modifiers:

```
T = assertion
P =
[[experiencer, [[entity, [i], reference(first)],
  [modification(attribute), quantity(single)]]],
 [interaction(experiencer_subj), [love]],
 [modification(attribute), time(present)],
 [content, [
  [interaction(ingestion), [eat]],
  [ingestibles, [entity, [food]]
  [modification(restriction), [mexican]],
 ]]],
 [reason, [[agent, [[entity, [it]],
  reference(third)],
  [modification(attribute), quantity(single)]]],
 [description,
 [modification(attribute), time(present)],
 [modification(attribute),
  taste_property(spicy)]]]
]
```

The corresponding parse tree is shown in Figure 1.

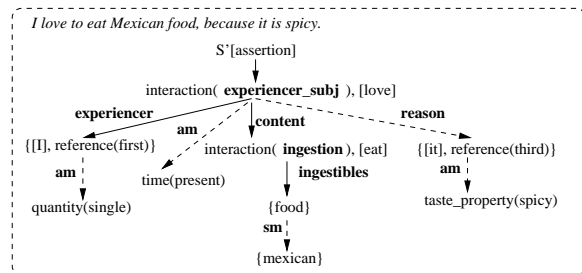


Figure 1: Semantic parse tree (*am* = attributive modifier, *rm* = referential modifier, *sm* = restrictive modifier)

We have conducted evaluations of the semantic role assignment algorithm on 350 sentences randomly selected from FrameNet. The test sentences were removed from the FrameNet corpus, and the rules-extraction procedure described earlier in the paper was invoked on this reduced corpus. All test sentences were then semantically parsed, and full semantic annotations were produced for each sentence. Notice that the evaluation is conducted only

<sup>7</sup>The semantic parser was demonstrated in a major Natural Language Processing conference, and can be also demonstrated during the workshop.

for semantic role assignment – since this is the only information available in FrameNet. The other semantic annotations produced by the parser (e.g. attribute, gender, countability) are not evaluated at this point, since there are no hand-validated annotations of this kind available in current resources.

Both frames and frame elements are automatically identified by the parser. Out of all the elements correctly identified, we found that 74.5% were assigned with the correct role (this is therefore the *accuracy* of role assignment), which compares favorably with previous results reported in the literature for this task. Notice also that since this is a rule-based approach, the parser does not need large amounts of annotated data, and it works well the same for words for which only one or two sentences are annotated.

## 6 Interface and Integration to Other Systems

The semantic algorithm uses linguistic knowledge, such as syntactic realization of semantic roles in a case frame, syntax-semantics mappings, and lexical semantic knowledge, to parse the semantic structure of open text. It can be regarded as a shallow semantic analyzer, which provides partial results for higher level understanding systems that can effectively utilize context, commonsense, and other types of knowledge, to achieve final accurate meaning interpretations, or use custom defined rules for high level processing in particular domains.

The matching/scoring scheme integrated in our algorithm can effectively identify the right semantic interpretation, but some semantic ambiguity cannot be resolved without enough context and commonsense knowledge. For example, although the famous meaningless sentence “colorless green ideas sleep furiously” can be correctly identified as semantically anomalous by the semantic parser, by analyzing the syntactic behavior of “sleep” and the selectional restrictions that we attach to this frame, the sentence “I saw the man in the park with the telescope” has several semantic interpretations. According to the commonsense knowledge that we encode in the semantic parser (mostly drawn from WordNet), *telescope* is defined as a tool to see something, and we may infer that “with telescope” in this sentence describes an instrument of “see”. However, without enough context, not even humans can rule out the possibility that the “telescope” is the man’s possession, rather than an instrument for the interaction “see”. The semantic parser maintains all possible interpretations that cannot be rejected by their syntactic and shallow semantic patterns, and



rank all of them by their scores as the likelihood of being the correct interpretation. Other systems can use high level knowledge such as common sense, context or user defined rules to choose the right interpretation.

As an integral part of the parsing system, we provide several interfaces that allow other systems or additional modules to change the behavior of the parser based on their rules and knowledge. One such interface is the *ontochg* predicate, which is called whenever the ontological category is identified for a constituent during the syntactic-semantic analysis. By default, it outputs the same ontological category as identified by the parser, but other systems can change the content of this predicate to replace the ontological category identified by the parser with other categories, according to their rules and knowledge. This is particularly useful for integrating add-ons capable of anaphora and metaphor resolution. The *adjatt* predicate is another interface for add-ons that can resolve polysemy of descriptive adjectives and adverbs. Due to polysemy, some descriptive adjectives and adverbs may modify different attributes in different situations and sometimes the resolution requires high level understanding using commonsense knowledge and context. These interfaces make the semantic parser more flexible, robust, and easier to integrate into other systems that achieve high level meaning processing and understanding.

## 7 Related Work

There are several statistical approaches for automatic semantic role labeling based on PropBank and FrameNet. (Gildea and Jurafsky, 2000) proposed a statistical approach based on FrameNet I data for annotation of semantic roles. Fleischman (Fleischman et al., 2003) used FrameNet annotations in a maximum entropy framework. A more flexible generative model is proposed in (Thompson et al., 2003), where *null-instantiated* roles can be also identified, and frames are not assumed to be known a-priori. These approaches exclusively focus on semantic roles labeling based on statistical methods, rather than analysis of the full structure of sentence semantics. However, a rule-based approach is closer to the way humans interpret the semantic structure of a sentence. Moreover, as mentioned earlier, the FrameNet data is not meant to be “statistically representative” (Johnson et al., 2002), but rather illustrative for various language constructs, and therefore a rule-based approach is more suitable for this lexical resource.

## 8 Conclusions

In this paper, we proposed an algorithm for open text shallow semantic parsing. The algorithm has the capability to analyze the semantic structure of a sentence, and show how the meaning of the entire sentence is composed of smaller semantic units, linked by various semantic relations. The parsing process utilizes linguistic knowledge, consisting of rules derived from a frame dataset (FrameNet), a semantic network (WordNet), as well as hand-coded rules of syntax-semantics mappings, which encode natural selectional restrictions. Parsing semantic structures allows semantic units and constituents to be accessed and processed in a more meaningful way than syntactic parsing, and enables higher-level text understanding applications. We believe that the semantic parser will prove useful for a range of language processing applications that require knowledge of text meaning, including word sense disambiguation, information retrieval, question answering, machine translation, and others.

## References

- M. Fleischman, N. Kwon, and E. Hovy. 2003. Maximum entropy models for FrameNet classification. In *Proceedings of 2003 Conference on Empirical Methods in Natural Language Processing EMNLP-2003*, Sapporo, Japan.
- D. Gildea and D. Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Conference of the Association for Computational Linguistics (ACL 2000)*, pages 512–520, Hong Kong, October.
- C. Johnson, C. Fillmore, M. Petruck, C. Baker, M. Ellsworth, J. Ruppenhofer, and E. Wood. 2002. *FrameNet: Theory and Practice*. <http://www.icsi.berkeley.edu/framenet>.
- K. Kipper, H.T.Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of Seventeenth National Conference on Artificial Intelligence AAAI 2000*, Austin, TX, July.
- B. Levin. 1993. *English Verb Classes and Alternation: A Preliminary Investigation*. The University of Chicago Press.
- G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.
- C. Thompson, R. Levy, and C. Manning. 2003. A generative model for FrameNet semantic role labeling. In *Proceedings of the Fourteenth European Conference on Machine Learning ECML-2003*, Croatia.