Semantic Attachments

Four core topics in Natural Language Processing

Semantic attachments are meaning-building operations associated with grammar productions. A parser uses them to combine word and phrase meanings into a formal sentence representation. These notes focus on syntax-driven compositional semantics.

Notation used throughout

S = sentence; NP = noun phrase; VP = verb phrase; PN = proper noun; V = verb. The attribute .sem stores meaning. An arrow indicates a grammar production; braces contain the semantic attachment. The symbol λ introduces a function, and applying that function substitutes an argument for its variable.

1. Attaching semantics to grammar rules

A grammar rule tells the parser how constituents combine. A semantic attachment adds an instruction for computing the meaning of the resulting constituent. Word meanings come from a semantic lexicon; phrase meanings are computed by attachments.

```
A → B C { A.sem = f(B.sem, C.sem) }
```

A is the parent constituent; B and C are its children. The function f combines their meanings. The attachment is part of the grammar specification, not part of the input sentence.

Example: “Ravi sleeps.”

Lexical meanings: Ravi = ravi; sleeps = λx.sleep(x).
Grammar: S → NP VP.
Attachment: S.sem = VP.sem(NP.sem).

```
S.sem = (λx.sleep(x))(ravi)
= sleep(ravi)
```

The NP names an individual. The VP supplies a property of individuals. Applying the property to ravi produces the proposition that Ravi sleeps.

Important distinction

Syntax builds the sentence structure; the attachment computes its interpretation. Creating sleep(ravi) represents the claim and does not establish its truth.

2. Rule-to-rule semantic mapping

The rule-to-rule approach pairs each relevant syntactic production with a semantic composition rule. Whenever a construction is recognized, its corresponding operation determines how its children's meanings combine. The operation must match the semantic types of the children.

Syntactic production	Semantic attachment	Meaning operation
NP → PN	NP.sem = PN.sem	Pass a name meaning upward.
VP → V NP	VP.sem = V.sem(NP.sem)	Apply the verb to its object.
S → NP VP	S.sem = VP.sem(NP.sem)	Apply the VP to its subject.

Why is it called rule-to-rule?

A rule for syntactic combination corresponds to a rule for semantic combination. This is a mapping from a grammatical construction to a meaning-building operation, rather than a word-by-word translation.

Lexical entries for the worked example

Ravi: ravi
Sita: sita
likes: λy.λx.like(x, y)

In like(x, y), x is the subject and y is the object. The verb takes its object first because V combines with the object NP to form a VP. The resulting VP then takes the subject.

How the mapping preserves roles

For "Ravi likes Sita," the VP rule supplies sita as the object and the S rule supplies ravi as the subject. The output is like(ravi, sita). For "Sita likes Ravi," the same rules produce like(sita, ravi).

Type compatibility

Proper names denote individuals. The VP denotes a function from an individual to a proposition. The S attachment therefore supplies an individual to a function that expects one. Incorrect function arguments can cause a semantic type mismatch.

Coverage and limitation

These rules form a miniature grammar for proper-name sentences. Quantified expressions such as "every student" require richer lexical meanings and compatible composition rules. A single attachment scheme does not automatically handle every English construction.

3. Semantic interpretation during parsing

An augmented parser stores semantic values along with syntactic information. When it recognizes a constituent and the required child meanings are available, it evaluates the associated attachment. Meaning construction can therefore proceed alongside parsing.

Worked example: “Ravi likes Sita.”

Step	Parser operation	Semantic value
1	Recognize PN “Ravi.”	PN.sem = ravi
2	Apply NP → PN to the subject.	Subject NP.sem = ravi
3	Recognize V “likes.”	V.sem = λy.λx.like(x, y)
4	Recognize PN “Sita” and form object NP.	Object NP.sem = sita
5	Apply VP → V NP.	VP.sem = λx.like(x, sita)
6	Apply S → NP VP.	S.sem = like(ravi, sita)

Step 5: compute the VP

```
VP.sem = V.sem(object NP.sem)
= (λy.λx.like(x, y))(sita)
= λx.like(x, sita)
```

Replace y with sita. The subject variable x remains unfilled, so the VP describes the property “likes Sita.”

Step 6: compute the sentence

```
S.sem = VP.sem(subject NP.sem)
= (λx.like(x, sita))(ravi)
= like(ravi, sita)
```

Replace x with ravi. Both arguments are filled, producing a complete proposition. These function applications and substitutions are called beta reductions.

Implementation choices

A bottom-up parser may evaluate an attachment when it reduces a production. A chart parser may store semantic information with chart items. Alternatively, a system can traverse a completed parse afterward. The exact timing depends on the parser; the attachment must have access to its required inputs.

If a sentence has several parses or word senses, the parser may retain several candidate meanings. The presence of attachments alone does not resolve ambiguity.

4. Role of attachments in meaning construction

Attachments connect grammatical structure to a usable meaning representation. They specify how local meanings are passed upward, applied to arguments and combined into a sentence-level result.

Role	Explanation	Example
Transfer lexical meaning	Make word meanings available to larger constituents.	NP.sem = PN.sem
Combine meanings	Construct a parent meaning from child meanings.	V.sem(sita)
Fill argument positions	Associate participants with the correct predicate roles.	Object = sita; subject = ravi
Build a formal result	Produce a logical form or another structured representation.	like(ravi, sita)
Support downstream use	Provide input for querying, reasoning or interpretation.	Check the proposition against a knowledge base.

Meaning-building path

Word meanings → noun-phrase meanings → verb-phrase meaning → sentence meaning. In the example, “likes” first combines with “Sita”; the resulting VP then combines with “Ravi.” The parse identifies the constituents, and the attachments specify the operations.

What attachments require from other components

They require a suitable grammar, lexical meanings and a parse. Resolving a pronoun such as “she,” selecting a word sense or identifying “the door” may require context. Establishing truth requires evidence or knowledge-base evaluation. Executing a command requires an application component.

Compact revision answer

Semantic attachments are operations attached to grammar rules. In rule-to-rule semantic mapping, syntactic productions are paired with meaning-composition rules. During augmented parsing, these operations use child meanings to compute parent meanings. Their role is to build a structured interpretation while preserving semantic roles. For “Ravi likes Sita,” the VP becomes λx.like(x, sita), and the sentence becomes like(ravi, sita).

Practice with answers

1. What does .sem represent? A constituent’s semantic value.
2. What remains open in λx.like(x, sita)? The subject argument x.
3. Interpret “Sita likes Ravi.” like(sita, ravi).
4. Does a logical form prove a statement true? No.

Reference

Daniel Jurafsky and James H. Martin, Speech and Language Processing, “Computational Semantics,” syntax-driven semantic analysis. Textbook PDF. Explanations are paraphrased; worked examples are illustrative. The small grammar omits tense, agreement and event structure.