

CFGs and Intro to Parsing

Scott Farrar
CLMA, University of Washington
farrar@uw.edu

January 11, 2010

Today's lecture

- 1 Practical Grammar Writing
 - Word classes
 - Clause/Phrase classes
 - Problems with the Treebank
 - Other notes about WSJ
- 2 Parsing: Key ideas
- 3 Approaches to parsing
 - Parsing Methods
 - Top-down parsing
 - Bottom-up parsing
- 4 Issues concerning natural language
 - Ambiguity
 - Recursion
 - Center embedding

Word classes and treebanks

Word classes

The number of word classes (pre-terminals) depends on the task and how fine you want to cut the pie (Tagged Brown corpus has 87 pre-terminal tags; Penn Treebank uses a 49-item pre-terminal tagset.) There's no right answer for NLP.

Penn Treebank has primarily been used for developing and testing parsers. A treebank or corpus used for semantic analysis or NLG might look very different.

A tour of the Penn Treebank and associated work:

Closed-class words

Definition

Closed class word: a function word in a grammar; there are relatively few of these in a language, though their frequency is very high. In treebank construction, such words can be, for the most part, tagged automatically.

Homework

This should be the easy part of hw1.

Closed classes

DT determiner
a(n), the, that, those

MD modal
do, can, may

PRP pronoun
she, her, him, he, we

EX existential there
there are many fish

CD cardinal number
one, two, three

...

(see list in front cover of J&M)

Open-class words

Definition

Open class word: a content word in a grammar; there is an open-ended set of these, but their frequencies may be very low (cf. *home* with *octogenarian*). Such words are harder to tag automatically in treebank construction. Why?

- Nouns
- Verbs
- Adjectives
- Adverbs

Nouns

Recall grade school definition:

Definition

A **noun** is a person, place, thing, or idea.

Nouns

Recall grade school definition:

Definition

A **noun** is a person, place, thing, or idea.

"You shall know a word by the company it keeps." J. R. Firth (d. 1960)

Nouns

Recall grade school definition:

Definition

A **noun** is a person, place, thing, or idea.

"You shall know a word by the company it keeps." J. R. Firth (d. 1960)

In other words, syntactic word categories are defined based on their **distribution**:

Nouns

Recall grade school definition:

Definition

A **noun** is a person, place, thing, or idea.

"You shall know a word by the company it keeps." J. R. Firth (d. 1960)

In other words, syntactic word categories are defined based on their **distribution**:

Definition

Noun is a class of lexical items that occur after determiners (*the*, *a*, ...) or adjectives, and can be subjects of sentences. Nouns often represent a person, place, thing, or idea.

Nouns

NN a singular common noun, occurring after adjectives and determiners

the [NN fisherman] caught it

Nouns

NN a singular common noun, occurring after adjectives and determiners

the [NN fisherman] caught it

NNS a plural common noun, occurring alone or after adjectives and determiners

[NNS fish] swim well

Nouns

NN a singular common noun, occurring after adjectives and determiners

the [NN fisherman] caught it

NNS a plural common noun, occurring alone or after adjectives and determiners

[NNS fish] swim well

NNP a proper noun or name, occurring alone in a noun phrase; does not (usually) occur after a determiner

[NNP Jack] knows

Nouns

NN a singular common noun, occurring after adjectives and determiners

the [NN fisherman] caught it

NNS a plural common noun, occurring alone or after adjectives and determiners

[NNS fish] swim well

NNP a proper noun or name, occurring alone in a noun phrase; does not (usually) occur after a determiner

[NNP Jack] knows

NNPS a plural proper noun

the [NNPS Simpsons] know the [NNP Jones]

Verbs

Definition

A **verb** describes states or events.

The forms of English verbs predict where they will occur. Consider these verb labels (based on WSJ corpus):

Verbs

Definition

A **verb** describes states or events.

The forms of English verbs predict where they will occur. Consider these verb labels (based on WSJ corpus):

VBD a past tense form occurs alone
the Earl [VBD ate] a sandwich

Verbs

Definition

A **verb** describes states or events.

The forms of English verbs predict where they will occur. Consider these verb labels (based on WSJ corpus):

VBD a past tense form occurs alone

the Earl [VBD ate] a sandwich

VBZ a third person form occurs after a singular (pro)noun

she [VBZ runs] two marathons a year

Verbs

Definition

A **verb** describes states or events.

The forms of English verbs predict where they will occur. Consider these verb labels (based on WSJ corpus):

VBD a past tense form occurs alone

the Earl [VBD ate] a sandwich

VBZ a third person form occurs after a singular (pro)noun

she [VBZ runs] two marathons a year

VBN a participle form occurs after *was, were, has, had,*

have, got, get, etc

he was [VBN bitten] by a tiger

Adjectives

Definition

Adjectives ascribe properties to nouns. They occur before nouns or after verbs in the predicate.

Adjectives

Definition

Adjectives ascribe properties to nouns. They occur before nouns or after verbs in the predicate.

- JJ a simple adjective
the [JJmetamorphic] rock,
the rock is [JJmetamorphic]

Adjectives

Definition

Adjectives ascribe properties to nouns. They occur before nouns or after verbs in the predicate.

JJ a simple adjective

the [JJmetamorphic] rock,

the rock is [JJmetamorphic]

JJR a comparative adjective

the [JJRbigger] rock

Adjectives

Definition

Adjectives ascribe properties to nouns. They occur before nouns or after verbs in the predicate.

JJ a simple adjective

the [JJmetamorphic] rock,

the rock is [JJmetamorphic]

JJR a comparative adjective

the [JJRbigger] rock

JJS a superlative adjective

the [JJSbiggest] one

Adverbs

Definition

Adverbs modify verbs (and adjectives) to specify time, manner, place, or direction of the event.

Adverbs

Definition

Adverbs modify verbs (and adjectives) to specify time, manner, place, or direction of the event.

RB an adverb can occur around the verb phrase or at the beginning/end of the clause (fast, quickly, really, here)

Adverbs

Definition

Adverbs modify verbs (and adjectives) to specify time, manner, place, or direction of the event.

RB an adverb can occur around the verb phrase or at the beginning/end of the clause (fast, quickly, really, here)

RBR comparative adverb: *ran* [**RBR***faster*] *than...*, *woke up* [**RBR***earlier*]

Adverbs

Definition

Adverbs modify verbs (and adjectives) to specify time, manner, place, or direction of the event.

- RB** an adverb can occur around the verb phrase or at the beginning/end of the clause (fast, quickly, really, here)
- RBR** comparative adverb: *ran* [*RBR* *faster*] *than...*, *woke up* [*RBR* *earlier*]
- RBS** superlative adverb: [*RBS* *most*] *notable*, *ran* [*RBS* *fastest*]

Other common abbreviations

Symbol	Meaning	Symbol	Meaning
Det	determiner	NP	noun phrase
Noun	noun	VP	verb phrase
Nom	nominal	AP	adjective phrase
Pro	pronoun	PP	prepositional phrase
Aux	auxiliary		
Card	cardinal number		
Ord	ordinal number		
Quant	quantifier		

Small grammar writing strategy

The task in grammar writing is to choose the best elements for nonterminals.

- 1 Settle on a tagset for pre-terminals (part-of-speech)
- 2 Tag data for part of speech
- 3 Identify larger clause patterns; come up with tags
- 4 Identify each phrase type; come up with tags
- 5 Fill in details for each phrase type
- 6 Identify major clause types
- 7 Address problematic cases

PTB phrase types

NP noun phrase including all constituents that depend on the noun head

VP : verb phrase including all constituents that depend on the verb head

PP : prepositional phrase

ADJP : adjective phrase headed by an adjective

ADVP : adverb phrase headed by an adverb

CONJP : used to mark multi-word conjunctions

QP : quantifier phrase, used inside *NPs*

• ...

PTB Clause types

The number of non-terminals (excluding pre-terminals) is generally small. In the Penn Treebank, there are, for example, 29 basic tags for syntactic constituents, including 5 basic clause types and 21 phrase-level constituents.

S declaratives, passives, imperatives, questions with declarative order, (embedded) infinitive clauses, gerund classes

SINV inverted clauses

SBAR relative and subordinate clauses

SBARQ Wh-questions

SQ Y/N-questions, inside *SBARQ*

S-CLF : it-cleft clauses

FRAG stand-alone clauses, phrases without a predicate argument structure.

Problems with Penn Treebank

As a CFG, why is the Penn Treebank fundamentally flawed?

Problems with Penn Treebank

As a CFG, why is the Penn Treebank fundamentally flawed?

- number of rules is intractably large 17,500, in order to parse 50,000 sentences

Problems with Penn Treebank

As a CFG, why is the Penn Treebank fundamentally flawed?

- number of rules is intractably large 17,500, in order to parse 50,000 sentences
- number of rules seems disproportionate at best

Problems with Penn Treebank

As a CFG, why is the Penn Treebank fundamentally flawed?

- number of rules is intractably large 17,500, in order to parse 50,000 sentences
- number of rules seems disproportionate at best
- number rules seems to grow linearly with the addition of new sentences

Problems with Penn Treebank

As a CFG, why is the Penn Treebank fundamentally flawed?

- number of rules is intractably large 17,500, in order to parse 50,000 sentences
- number of rules seems disproportionate at best
- number rules seems to grow linearly with the addition of new sentences

Main point

The rules do not express linguistic generalizations.

Rule growth in the Penn Treebank

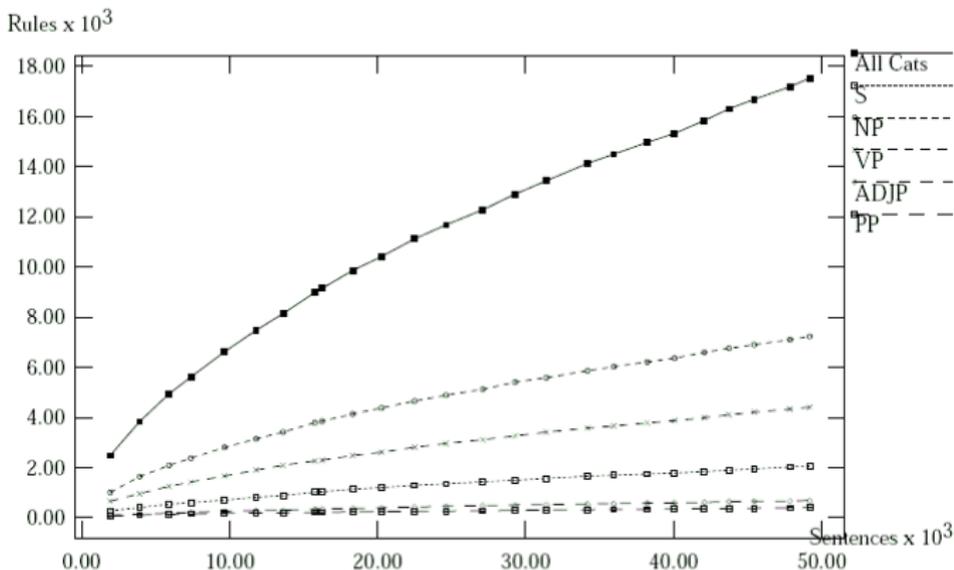


Figure 1: Rate of Growth of Rule Set Size

Problems w the Treebank

Why is the rule set so large?

Problems w the Treebank

Why is the rule set so large?

- diversity of language

Problems w the Treebank

Why is the rule set so large?

- diversity of language
- some sort of generative process going on (in the heads of annotators)

Problems w the Treebank

Why is the rule set so large?

- diversity of language
- some sort of generative process going on (in the heads of annotators)
- shallow analysis of sentence by annotators

Some Solutions

See Gaizaukas paper

Some Solutions

See Gaizaukas paper

- eliminate low frequency rules

Some Solutions

See Gaizaukas paper

- eliminate low frequency rules
- 2144 rules account for 95% of grammar

Some Solutions

See Gaizaukas paper

- eliminate low frequency rules
- 2144 rules account for 95% of grammar
- author used 100 rules to obtain a grammar that accounted for 70% of rule occurrences

Some Solutions

See Gaizaukas paper

- eliminate low frequency rules
- 2144 rules account for 95% of grammar
- author used 100 rules to obtain a grammar that accounted for 70% of rule occurrences
- try to parse RHS of low frequency rules with higher ones

Some Solutions

See Gaizaukas paper

- eliminate low frequency rules
- 2144 rules account for 95% of grammar
- author used 100 rules to obtain a grammar that accounted for 70% of rule occurrences
- try to parse RHS of low frequency rules with higher ones

Goal

Come up with a tractable, yet expressive grammar for parsing experiments.

The Penn Treebank parses

What are you thinking about?

(SBARQ

(WHNP (WP What))

(SQ (VBP are)

(NP (PRP you))

(VP

(VBG thinking)

(IN about)))

(PUNC ?))

Traces in the Penn Treebank

What are you thinking about **T**?

Traces in the Penn Treebank

What are you thinking about *T*?

```
(SBARQ
  (WHNP (WP What))
  (SQ (VBP are)
      (NP (PRP you))
      (VP (VBG thinking)
          (PP (IN about)
              (NP *T*))))))
(PUNC ?))
```

Traces in the Penn Treebank

Where did I put the marker?

```
(SBARQ
  (WHADVP (WRB Where))
  (SQ (VBD did)
    (NP (PRP I))
    (VP (VB put)
      (NP
        (DT the)
        (NN marker))))
  (PUNC ?))
```

Traces in the Penn Treebank

Where did I put the marker *T*?

```
(SBARQ
  (WHADVP (WRB Where))
  (SQ (VBD did)
    (NP (PRP I))
    (VP (VB put)
      (NP
        (DT the)
        (NN marker)
        (ADVP *T*))))
  (PUNC ?))
```

Parsing

Definition

Parsing is the task of deriving a **structural description** of natural language utterances. Given a sentence S of natural language and some grammar G , the parsing task is to return a syntactic structure, in the form of a parse-tree T , of S .

Parsing

Definition

Parsing is the task of deriving a **structural description** of natural language utterances. Given a sentence S of natural language and some grammar G , the parsing task is to return a syntactic structure, in the form of a parse-tree T , of S .

Definition

A variant of parsing is **recognition**: Given a sentence S of natural language and some grammar G , the recognition task is to return **true**, if S is a valid sentence of G —i.e., if a syntactic structure can be found—or **false** otherwise.

Parsing

Why parse?

Parsing is used for: grammar checking, speech recognition, deriving a semantic representation (for MT, question-answering, information extraction), and many other NLP tasks.

Parsing

Why parse?

Parsing is used for: grammar checking, speech recognition, deriving a semantic representation (for MT, question-answering, information extraction), and many other NLP tasks.

It's all about getting at the units, or parts (parse from Lt. *pars*)

Parsing

Why parse?

Parsing is used for: grammar checking, speech recognition, deriving a semantic representation (for MT, question-answering, information extraction), and many other NLP tasks.

It's all about getting at the units, or parts (parse from Lt. *pars*)

Orthographic (or phonological) units will ultimately reveal patterns that map onto the semantic units (according to the grammar). Those patterns, in some sense, are the syntax of the language (recall definition).

Parser Demo

There are several parser available here:

`/NLP_TOOLS/parsers`

```
$ cd ~/dropbox/09-10/571/misc_code/stanford_parser
```

```
$ ./parse
```

Parsing as search

The parsing task can be approached as a **search problem**.

Parsing as search

The parsing task can be approached as a **search problem**.

Definition

A **search algorithm** is one that starts with a problem input and returns a number of solutions based on some method of generating the possible solutions.

A quick overview of search

Elements of search

Search can be conceptualized as a tree of partial to complete solutions:

A quick overview of search

Elements of search

Search can be conceptualized as a tree of partial to complete solutions:

- **tree search:** a strategy that generates a tree of possible solutions.

A quick overview of search

Elements of search

Search can be conceptualized as a tree of partial to complete solutions:

- **tree search:** a strategy that generates a tree of possible solutions.
- **search node:** a data structure holding information about some step in the solution process.

A quick overview of search

Elements of search

Search can be conceptualized as a tree of partial to complete solutions:

- **tree search**: a strategy that generates a tree of possible solutions.
- **search node**: a data structure holding information about some step in the solution process.
- **solution node**: a search node containing a solution.

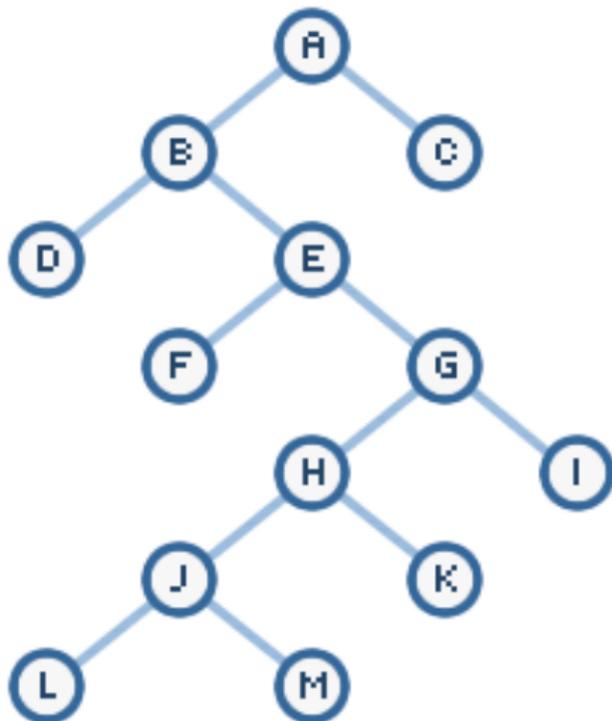
A quick overview of search

Elements of search

Search can be conceptualized as a tree of partial to complete solutions:

- **tree search:** a strategy that generates a tree of possible solutions.
- **search node:** a data structure holding information about some step in the solution process.
- **solution node:** a search node containing a solution.
- **search space:** the set of all possible solutions (including solution paths) to a search problem

Search example



Searching for a parse

- **search node:** a partial parse tree
the cat (PP (IN in ((DT the) (NN hat))))
- **solution node:** a complete parse tree
- **search space:** all the paths that lead to a successful parse
and all the dead-ends

Elements of search

How to expand each node? And how do we determine success?

Elements of search

How to expand each node? And how do we determine success?

- **expansion function:** a way to build the contents of the next node and expand the search tree.

Elements of search

How to expand each node? And how do we determine success?

- **expansion function:** a way to build the contents of the next node and expand the search tree.
- **evaluation function:** one that returns true if a solution is found at a solution node.

Varying the search strategies

Exploring the space

Two ways to explore the search space:

Varying the search strategies

Exploring the space

Two ways to explore the search space:

- 1 **Breadth-first search** is an uninformed search strategy whereby the search space is explored by visiting all neighboring (sister) nodes first, before going deeper into the tree.

Varying the search strategies

Exploring the space

Two ways to explore the search space:

- 1 **Breadth-first search** is an uninformed search strategy whereby the search space is explored by visiting all neighboring (sister) nodes first, before going deeper into the tree.
- 2 **Depth-first search** is an uninformed search strategy whereby the search space is explored by going deeper and deeper (down a branch of the tree structure) until backtracking is required.

Varying the expansion strategy

For NL parsing the choice of expansion function is important:

- 1 top-down parse tree expansion
- 2 bottom-up parse tree expansion

Top-down parsing

Definition

Using a **top-down** parse tree expansion strategy, start with the root node (e.g. S) and work towards the solution via subgoals, namely solutions for NP , VP , etc. In other words, starting with the root node of the parse tree, progress towards the goal, which is the full parse tree, by progressively expanding the parse tree.

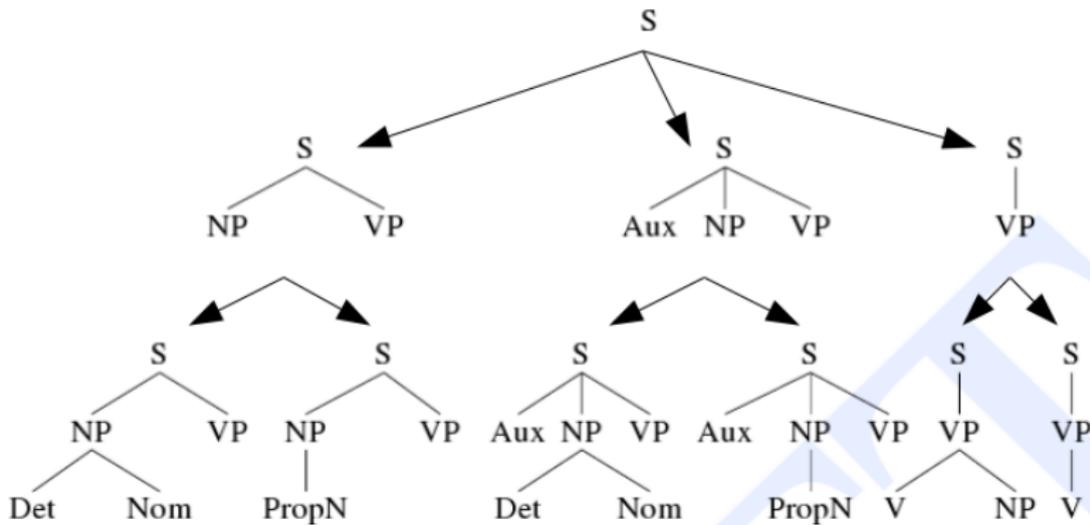
Top-down parsing

Definition

Using a **top-down** parse tree expansion strategy, start with the root node (e.g. S) and work towards the solution via subgoals, namely solutions for NP , VP , etc. In other words, starting with the root node of the parse tree, progress towards the goal, which is the full parse tree, by progressively expanding the parse tree.

An example of a top-down parser is the **recursive descent** parser which tries to build a tree (top-down) by iterating over the rules of the grammar. It backtracks when no terminal is matched.

Top-down parse example



Pros/cons of top-down strategy

- ✓ Never explores trees that aren't potential solutions, ones with the wrong kind of root node.

Pros/cons of top-down strategy

- ✓ Never explores trees that aren't potential solutions, ones with the wrong kind of root node.
- ✗ But explores trees that do not match the input sentence (predicts input before inspecting input).

Pros/cons of top-down strategy

- ✓ Never explores trees that aren't potential solutions, ones with the wrong kind of root node.
- ✗ But explores trees that do not match the input sentence (predicts input before inspecting input).
- ✗ Naive top-down parsers never terminate if G contains recursive rules like $X \rightarrow X Y$ (left recursive rules).

Pros/cons of top-down strategy

- ✓ Never explores trees that aren't potential solutions, ones with the wrong kind of root node.
- ✗ But explores trees that do not match the input sentence (predicts input before inspecting input).
- ✗ Naive top-down parsers never terminate if G contains recursive rules like $X \rightarrow X Y$ (left recursive rules).
- ✗ Backtracking may discard valid constituents that have to be re-discovered later (duplication of effort).

Pros/cons of top-down strategy

- ✓ Never explores trees that aren't potential solutions, ones with the wrong kind of root node.
- ✗ But explores trees that do not match the input sentence (predicts input before inspecting input).
- ✗ Naive top-down parsers never terminate if G contains recursive rules like $X \rightarrow X Y$ (left recursive rules).
- ✗ Backtracking may discard valid constituents that have to be re-discovered later (duplication of effort).

Use a top-down strategy when you know what kind of constituent you want to end up with (e.g. NP extraction, named entity extraction). Avoid this strategy if you're stuck with a highly recursive grammar.

Bottom-up parsing

Definition

Using a **bottom-up** parse tree expansion strategy, starting with the sentence, progress towards the goal, i.e., the full parse tree, by progressively building the parse tree. In other words, try to match the right-hand side of rules to build a partial solution, progressively building structure upwards.

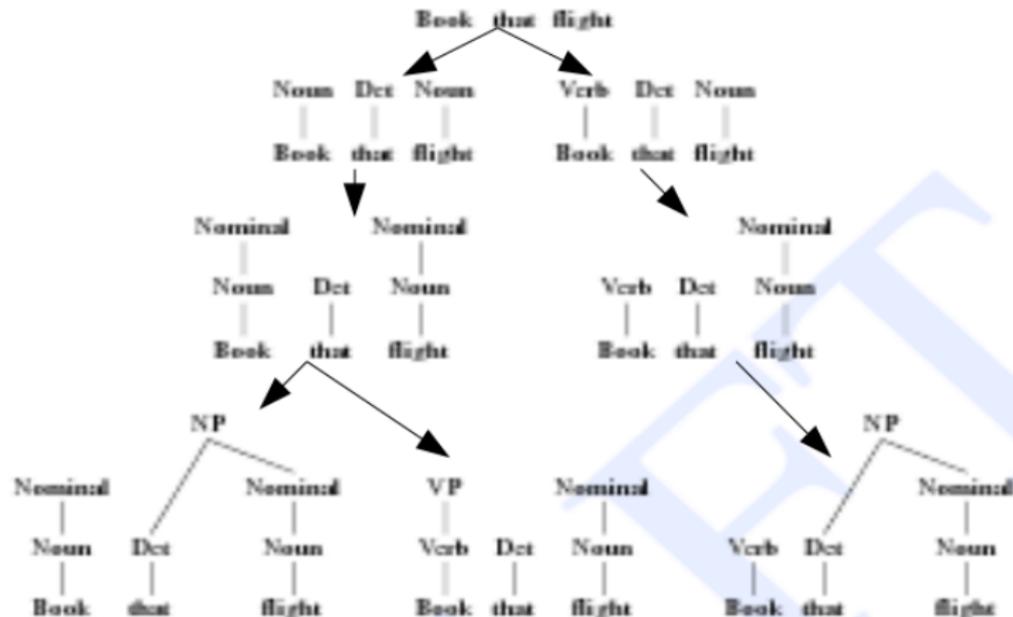
Bottom-up parsing

Definition

Using a **bottom-up** parse tree expansion strategy, starting with the sentence, progress towards the goal, i.e., the full parse tree, by progressively building the parse tree. In other words, try to match the right-hand side of rules to build a partial solution, progressively building structure upwards.

An example is the **shift-reduce** parser. Push input words onto a stack (shift) and try to build structure (reduce).

Bottom-up parse example



Pros/cons of bottom-up strategy

Pros/cons of bottom-up strategy

- ✓ Locally grounded in the input sentence.

Pros/cons of bottom-up strategy

- ✓ Locally grounded in the input sentence.
- ✓ Recursive rules are not generally a problem.

Pros/cons of bottom-up strategy

- ✓ Locally grounded in the input sentence.
- ✓ Recursive rules are not generally a problem.
- ✓ Substructures are only built once.

Pros/cons of bottom-up strategy

- ✓ Locally grounded in the input sentence.
- ✓ Recursive rules are not generally a problem.
- ✓ Substructures are only built once.
- ✗ Explores many trees that are not rooted with goal nodes.
(Shift-reduce algorithm can fail to find any parse.)

Pros/cons of bottom-up strategy

- ✓ Locally grounded in the input sentence.
- ✓ Recursive rules are not generally a problem.
- ✓ Substructures are only built once.
- ✗ Explores many trees that are not rooted with goal nodes.
(Shift-reduce algorithm can fail to find any parse.)

Use this type of parser when you're parsing real-time speech input.
Why?

Difficulties in parsing NL

Difficulties in parsing NL

- **Ambiguity**: more than one solution (more than one structural description)

Difficulties in parsing NL

- **Ambiguity**: more than one solution (more than one structural description)
- **Recursion**: production rules whose RHS contains the LHS symbol (e.g, $S \rightarrow S \text{ CONJ } S$)

Difficulties in parsing NL

- **Ambiguity**: more than one solution (more than one structural description)
- **Recursion**: production rules whose RHS contains the LHS symbol (e.g, $S \rightarrow S \text{ CONJ } S$)
- **Center embedding**: structure within structure
The cat [that sat in the chair under the lamp beside the couch] licked its paws

Ambiguity in natural language

Ambiguous input poses problems for parsers.

Ambiguity in natural language

Ambiguous input poses problems for parsers.

- Book that flight.

Ambiguity in natural language

Ambiguous input poses problems for parsers.

- Book that flight.
- Time flies like an arrow.

Ambiguity in natural language

Ambiguous input poses problems for parsers.

- Book that flight.
- Time flies like an arrow.
- Canadian history teacher

Ambiguity in natural language

Ambiguous input poses problems for parsers.

- Book that flight.
- Time flies like an arrow.
- Canadian history teacher
- Galileo saw Medici's wife with a telescope.

Ambiguity in natural language

Ambiguous input poses problems for parsers.

- Book that flight.
- Time flies like an arrow.
- Canadian history teacher
- Galileo saw Medici's wife with a telescope.
- I ran with my dog.

Types of ambiguity

Two types of ambiguity most relevant for parsing:

Types of ambiguity

Two types of ambiguity most relevant for parsing:

- **lexical ambiguity**: uncertainty introduced when a word token belongs to more than one part-of-speech category.
house/NN, house/VB
sweet/JJ, sweet/NN.

Types of ambiguity

Two types of ambiguity most relevant for parsing:

- **lexical ambiguity**: uncertainty introduced when a word token belongs to more than one part-of-speech category.
house/NN, house/VB
sweet/JJ, sweet/NN.
- **structural ambiguity**: uncertainty introduced by having more than one rule that can describe a given string: a string has more than one structure.

Structural ambiguity

Two types of structural ambiguity:

Structural ambiguity

Two types of structural ambiguity:

- **attachment ambiguity**: when a constituent can be attached at more than place in the parse tree

Structural ambiguity

Two types of structural ambiguity:

- **attachment ambiguity**: when a constituent can be attached at more than place in the parse tree
- **coordination ambiguity**: when different constituents can be formed from a conjunction (*and*, *or*)

Structural ambiguity

Two types of structural ambiguity:

- **attachment ambiguity**: when a constituent can be attached at more than place in the parse tree
- **coordination ambiguity**: when different constituents can be formed from a conjunction (*and*, *or*)

Parsers that find all possible parses for a given input must be able to **disambiguate** and choose one candidate.

Recursion in NL

Definition

Recursion: A process in a grammatical derivation whereby a rule is re-applied to itself, resulting in the same pattern being repeated over and over: $[_{Nom} X[_{Nom} Y[_{PP} Z]]]$

Recursion in NL

Definition

Recursion: A process in a grammatical derivation whereby a rule is re-applied to itself, resulting in the same pattern being repeated over and over: $[_{Nom} X[_{Nom} Y[_{PP} Z]]]$

- Direct recursion:

$Nom \rightarrow Nom PP, S \rightarrow S CONJ S$

water under the bridge, Bill ran and Jane jogged

Recursion in NL

Definition

Recursion: A process in a grammatical derivation whereby a rule is re-applied to itself, resulting in the same pattern being repeated over and over: $[_{Nom} X[_{Nom} Y[_{PP} Z]]]$

- Direct recursion:

$Nom \rightarrow Nom PP, S \rightarrow S CONJ S$

water under the bridge, Bill ran and Jane jogged

- Indirect recursion

*... on the thimble in the box on the stool beside the table
near the sofa ...*

$NP \rightarrow DT Nom$

$Nom \rightarrow Nom PP$

$PP \rightarrow Prep NP$

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

- The company failed.

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

- The company failed.
- The company the law firm sued failed.

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

- The company failed.
- The company the law firm sued failed.
- The company [the law firm sued] failed.

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

- The company failed.
- The company the law firm sued failed.
- The company [the law firm sued] failed.
- The company the law firm the boss hired sued failed.

Center embedding

Definition

Center embedding: When a syntactic constituent A is contained/nested within another constituent B and surrounded by other constituents X and Z : $[_B X [_A Y] Z]$

- The company failed.
- The company the law firm sued failed.
- The company [the law firm sued] failed.
- The company the law firm the boss hired sued failed.
- The company [the law firm [the boss hired] sued]failed.

Garden path sentences

- The horse raced past the barn fell.

Garden path sentences

- The horse raced past the barn fell.
- The horse [raced past the barn] fell.

Garden path sentences

- The horse raced past the barn fell.
- The horse [raced past the barn] fell.
- The horse which was raced past the barn fell.

Garden path sentences

- The horse raced past the barn fell.
- The horse [raced past the barn] fell.
- The horse which was raced past the barn fell.
- The mayor forced out of office was arrested.

Garden path sentences

- The horse raced past the barn fell.
- The horse [raced past the barn] fell.
- The horse which was raced past the barn fell.
- The mayor forced out of office was arrested.

Definition

Garden path sentences are those for which unnecessary structure is built up during the parsing process. The parser is then forced to 'undo' the structure already built. These pose particular problems for human sentence processing.