Grammar

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

January 3, 2010

Today's lecture

[continued from Monday's lecture]

What is syntax?

Definition

Syntax is the study of how the parts of an utterance are arranged in relation to one another.

What is syntax?

Definition

Syntax is the study of how the parts of an utterance are arranged in relation to one another.

Some questions for syntax:

- Do all languages behave the same way?

What is syntax?

Definition

Syntax is the study of how the parts of an utterance are arranged in relation to one another.

Some questions for syntax:

- Do all languages behave the same way?
- Can the structure of yet un-analyzed languages be predicted?

What is syntax?

Definition

Syntax is the study of how the parts of an utterance are arranged in relation to one another.

Some questions for syntax:

- Do all languages behave the same way?
- Can the structure of yet un-analyzed languages be predicted?
- How is syntax learned by children (with little negative evidence)?

What is syntax?

Definition

Syntax is the study of how the parts of an utterance are arranged in relation to one another.

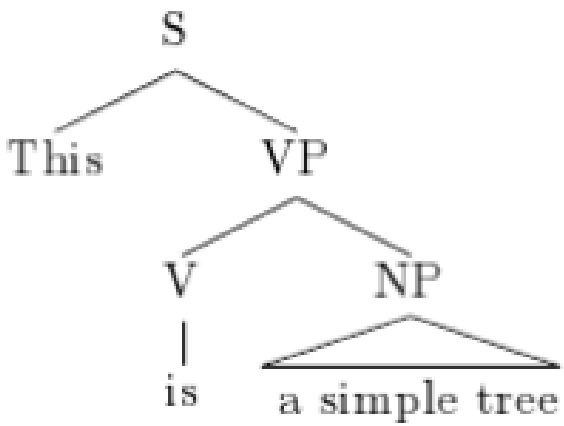
Some questions for syntax:

- Do all languages behave the same way?
- Can the structure of yet un-analyzed languages be predicted?
- How is syntax learned by children (with little negative evidence)?

Definition

A basic construct of syntax is the structural description, a structure that shows word order, syntactic constituency, and labels for the constituents.

Structural description: tree



Structural description: bracketed structure

[This [is [a simple bracketed structure]]]

...with no labels.

Structural description: bracketed structure

[This [is [a simple bracketed structure]]]

...with no labels.

[$_S$ *This*[$_{VP}$[$_V$ *is*][$_{NP}$ *a simple bracketed structure*]]]

The labeled bracketed structure is equivalent to the tree structure.

Tree structures for linguistics

Definition

A tree is a graphical way of representing structural description; trees for NL are more precisely *ordered directed trees* with (nodes, labels and arcs).

Node	a component or unit of a tree.
Root node	the node with no ancestors (labeled by the start symbol.
Nonterminal node	a node with descendants.
Terminal/Leaf node	a node with no descendants (corresponding to the strings, or "words", of the language).

Tree structures for linguistics

Definition

A tree is a graphical way of representing structural description; trees for NL are more precisely *ordered directed trees* with (nodes, labels and arcs).

Node a component or unit of a tree.

Root node the node with no ancestors (labeled by the start symbol.

Nonterminal node a node with descendants.

Terminal/Leaf node a node with no descendants (corresponding to the strings, or "words", of the language).

Tree structures for linguistics

Definition

A tree is a graphical way of representing structural description; trees for NL are more precisely *ordered directed trees* with (nodes, labels and arcs).

Node	a component or unit of a tree.
Root node	the node with no ancestors (labeled by the start symbol.
Nonterminal node	a node with descendants.
Terminal/Leaf node	a node with no descendants (corresponding to the strings, or "words", of the language).

Tree structures for linguistics

Definition

A tree is a graphical way of representing structural description; trees for NL are more precisely *ordered directed trees* with (nodes, labels and arcs).

Node	a component or unit of a tree.
Root node	the node with no ancestors (labeled by the start symbol.
Nonterminal node	a node with descendants.
Terminal/Leaf node	a node with no descendants (corresponding to the strings, or "words", of the language).

Tree structures for linguistics

Definition

A tree is a graphical way of representing structural description; trees for NL are more precisely *ordered directed trees* with (nodes, labels and arcs).

Node	a component or unit of a tree.
Root node	the node with no ancestors (labeled by the start symbol.
Nonterminal node	a node with descendants.
Terminal/Leaf node	a node with no descendants (corresponding to the strings, or "words", of the language).

Tree structures for linguistics

Preterminal node the node with only a single leaf as its descendant.

In NL grammar, these are the part of speech nodes.

Tree structures for linguistics

Preterminal node the node with only a single leaf as its descendant.
In NL grammar, these are the part of speech nodes.

Arc shows the constituency relation, but is untyped.

Tree structures for linguistics

Preterminal node the node with only a single leaf as its descendant.
In NL grammar, these are the part of speech nodes.

Arc shows the constituency relation, but is untyped.

Label a symbol identifying the category of some node.

What is syntax?

Definition

In another sense, syntax is the set of rules by which well formed utterances are formed. (The term *grammar* is more general and refers to all aspects of language.)

What is syntax?

Definition

In another sense, syntax is the set of rules by which well formed utterances are formed. (The term *grammar* is more general and refers to all aspects of language.)

We can formalize the notion of syntax using ideas from **formal language theory**.

Formal language theory

Since natural language is rule-governed, not random, a grammar can be constructed to parse natural language, just as with compilers and machine languages.

Definition

Formal language theory, or the study of the properties of formal languages, gives a firm conceptual framework from which to study natural language.

Chomsky hierarchy

Definition

The Chomsky Hierarchy describes four classes of formal grammars that generate four corresponding classes of languages.

(We only talk about CFGs in Ling571, but see J&M Chap. 16 for an overview.)

Chomsky hierarchy

Definition

The Chomsky Hierarchy describes four classes of formal grammars that generate four corresponding classes of languages.

1. **Type 0**: Turing equivalent language/grammar

(We only talk about CFGs in Ling571, but see J&M Chap. 16 for an overview.)

Chomsky hierarchy

Definition

The Chomsky Hierarchy describes four classes of formal grammars that generate four corresponding classes of languages.

1. **Type 0**: Turing equivalent language/grammar
2. **Type 1**: context sensitive language/grammar

(We only talk about CFGs in Ling571, but see J&M Chap. 16 for an overview.)

Chomsky hierarchy

Definition

The Chomsky Hierarchy describes four classes of formal grammars that generate four corresponding classes of languages.

1. **Type 0**: Turing equivalent language/grammar
2. **Type 1**: context sensitive language/grammar
3. Type 2: context free language/grammar

(We only talk about CFGs in Ling571, but see J&M Chap. 16 for an overview.)

Chomsky hierarchy

Definition

The Chomsky Hierarchy describes four classes of formal grammars that generate four corresponding classes of languages.

1. **Type 0**: Turing equivalent language/grammar
2. **Type 1**: context sensitive language/grammar
3. Type 2: context free language/grammar
4. **Type 3**: regular language/grammar

(We only talk about CFGs in Ling571, but see J&M Chap. 16 for an overview.)

Formal grammar

Definition

(Generalized) formal grammar A grammar is defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- Σ is a set of terminals, typically $x, y, z, \ldots$
- P is a set of production rules
- S is the starting or goal variable from N, i.e., $S \in N$

Formal grammar

Definition

(Generalized) formal grammar A grammar is defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- Σ is a set of terminals, typically $x, y, z, \ldots$
- P is a set of production rules
- S is the starting or goal variable from N, i.e., $S \in N$

Formal grammar

Definition

(Generalized) formal grammar A grammar is defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- Σ is a set of terminals, typically $x, y, z, \ldots$
- P is a set of production rules
- S is the starting or goal variable from N, i.e., $S \in N$

Formal grammar

Definition

(Generalized) formal grammar A grammar is defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- Σ is a set of terminals, typically $x, y, z, \ldots$
- P is a set of production rules
- S is the starting or goal variable from N, i.e., $S \in N$

Formal grammar

Definition

(Generalized) formal grammar A grammar is defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- Σ is a set of terminals, typically $x, y, z, \ldots$
- P is a set of production rules
- S is the starting or goal variable from N, i.e., $S \in N$

Sample grammar

```
S -> NP VP
NP -> Det Noun
NP -> ProperNoun
VP -> Verb
VP -> Verb NP

Det -> the|a|that
Noun -> lamp|pig|dirt
ProperNoun -> Washington|Sam
Verb -> understands|chases
```

Washington understands Sam
Sam chases that pig
*understands Sam

Sample grammar

```
S -> NP VP
NP -> Det Noun
NP -> ProperNoun
VP -> Verb
VP -> Verb NP

Det -> the|a|that
Noun -> lamp|pig|dirt
ProperNoun -> Washington|Sam
Verb -> understands|chases
```

Washington understands Sam

Sam chases that pig

*understands Sam

Sample grammar

```
S -> NP VP
NP -> Det Noun
NP -> ProperNoun
VP -> Verb
VP -> Verb NP

Det -> the|a|that
Noun -> lamp|pig|dirt
ProperNoun -> Washington|Sam
Verb -> understands|chases
```

Washington understands Sam

Sam chases that pig

*understands Sam

Sample grammar

```
S -> NP VP
NP -> Det Noun
NP -> ProperNoun
VP -> Verb
VP -> Verb NP

Det -> the|a|that
Noun -> lamp|pig|dirt
ProperNoun -> Washington|Sam
Verb -> understands|chases
```

Washington understands Sam

Sam chases that pig

*understands Sam

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \longrightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Context-free grammar

Definition

A CFG grammar is formally defined as $G = \langle N, \Sigma, P, S \rangle$ where:

- N is a set of non-terminal symbols, typically $S, A, B, \ldots$
- S is the starting or goal symbol from N, i.e., $S \in N$
- Σ is a set of terminal symbols, typically $x, y, z, \ldots$ disjoint from N
- P is a set of production rules
- The productions P are of the form: $A \rightarrow \beta$, where:
  - A is a non-terminal $A \in N$
  - β is a string of symbols from $(\Sigma \cup N)$

Unpacking the definition

- A nonterminal symbol labels a syntactic part (constituent): *NP*, *VP*, *PP*, (*Noun*, *Verb*, *Det*)
- A starting symbol indicates which symbol has to come first; it labels the largest constituent or biggest part: *S*, *Root*, or *Top*
- A terminal symbol labels the smallest part, the actual strings of the language: *man*, *they*, *swim*

Unpacking the definition

- A nonterminal symbol labels a syntactic part (constituent): *NP*, *VP*, *PP*, (*Noun*, *Verb*, *Det*)
- A starting symbol indicates which symbol has to come first; it labels the largest constituent or biggest part: *S*, *Root*, or *Top*
- A terminal symbol labels the smallest part, the actual strings of the language: *man*, *they*, *swim*

Unpacking the definition

- A nonterminal symbol labels a syntactic part (constituent):
  NP, *VP*, *PP*, (*Noun*, *Verb*, *Det*)
- A starting symbol indicates which symbol has to come first; it labels the largest constituent or biggest part:
  S, *Root*, or *Top*
- A terminal symbol labels the smallest part, the actual strings of the language:
  man, *they*, *swim*

Unpacking the definition

- A production rule (a.k.a. re-write rule) indicates when one symbol is to be rewritten ($\rightarrow$) as one or more others: $NP \rightarrow Det\ Noun$
- The resulting symbols are, thus, derived from the parent.
- A production rule captures the notion of syntactic constituency. 'LHS' is used to indicate the left-hand side of the $\rightarrow$, and likewise for 'RHS'.

Unpacking the definition

- A production rule (a.k.a. re-write rule) indicates when one symbol is to be rewritten ($\rightarrow$) as one or more others: $NP \rightarrow Det\ Noun$
- The resulting symbols are, thus, derived from the parent.
- A production rule captures the notion of syntactic constituency. 'LHS' is used to indicate the left-hand side of the $\rightarrow$, and likewise for 'RHS'.

Unpacking the definition

- A production rule (a.k.a. re-write rule) indicates when one symbol is to be rewritten ($\rightarrow$) as one or more others: $NP \rightarrow Det\ Noun$
- The resulting symbols are, thus, derived from the parent.
- A production rule captures the notion of syntactic constituency. 'LHS' is used to indicate the left-hand side of the $\rightarrow$, and likewise for 'RHS'.

Is this a valid CFG grammar?

```
NP -> (Det) Nom
Nom -> (Adj) Noun
VP -> VB NP

Det -> the | a
Noun -> colonel | chicken
Adj -> fried | baked
VB -> ate | likes

the colonel ate fried chicken
```

Is this a valid CFG grammar?

```
NP -> (Det) Nom
Nom -> (Adj) Noun
VP -> VB NP

Det -> the | a
Noun -> colonel | chicken
Adj -> fried | baked
VB -> ate | likes

the colonel ate fried chicken
```

Needs a start symbol and associated rule: $S \rightarrow NP\ VP$

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.
Examples of treebanks:

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.

Examples of treebanks:

- The **Penn Treebank** is a tagged, parsed corpus of English (most well-known treebank)

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.
Examples of treebanks:

- The **Penn Treebank** is a tagged, parsed corpus of English (most well-known treebank)
- Penn Chinese Treebank Project

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.

Examples of treebanks:

- The **Penn Treebank** is a tagged, parsed corpus of English (most well-known treebank)
- Penn Chinese Treebank Project
- The Tübingen Treebank of Written German

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.

Examples of treebanks:

- The **Penn Treebank** is a tagged, parsed corpus of English (most well-known treebank)
- Penn Chinese Treebank Project
- The Tübingen Treebank of Written German
- Arabic Treebank

Grammars and treebanks

Definition

A linguistic corpus is a collection of naturally occurring human language. A treebank is a linguistic corpus annotated for syntactic structure.

Thus, a treebank implicitly contains a grammar (grammars) of the language it contains.

Examples of treebanks:

- The **Penn Treebank** is a tagged, parsed corpus of English (most well-known treebank)
- Penn Chinese Treebank Project
- The Tübingen Treebank of Written German
- Arabic Treebank
- Korean Treebank

Corpora on Patas

```
stiv@patas:/corpora$ ls
birkbeck  Conll europarl-old  JRC-Acquis.3.0  nltk
treebanks coconut europarl ICAME   LDC  TREC  wordnet
```

Corpora on Patas

```
stiv@patas:/corpora$ ls
birkbeck  Conll europarl-old  JRC-Acquis.3.0  nltk
treebanks coconut europarl ICAME   LDC  TREC  wordnet
```

- Largest collection is the Linguistics Data Consortium (LDC) corpora.
- The NLTK comes with many corpora fragments as well.
- See the compling database to search for specific corpora: https://pongo.ling.washington.edu/db/index.php
  Perform a search for 'Treebank'

Penn Treebank (2 and 3)

```
cd /corpora/LDC/LDC99T42/RAW

(see readme.1st)

cd /corpora/LDC/LDC99T42/RAW/parsed/mrg/wsj/04$

(see wsj_0432.mrg)
```

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.

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.

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*).

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 $[_{NNP}$*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.

Adjectives

Definition

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

JJ a simple adjective
the [$_{JJ}$*metamorphic*] *rock*,
the rock is [$_{JJ}$*metamorphic*]

Adjectives

Definition

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

JJ a simple adjective
the [$_{JJ}$ *metamorphic*] *rock*,
the rock is [$_{JJ}$ *metamorphic*]

JJR a comparative adjective
the [$_{JJR}$ *bigger*] *rock*

Adjectives

Definition

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

JJ a simple adjective
the [$_{JJ}$ *metamorphic*] *rock*,
the rock is [$_{JJ}$ *metamorphic*]

JJR a comparative adjective
the [$_{JJR}$ *bigger*] *rock*

JJS a superlative adjective
the [$_{JJS}$ *biggest*] *one*

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,

Other categories

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)

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		

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

Rules

Practical treebanks have a large number of rules. The Penn Treebank has more than 17,000! rules. Most of them are flat, and tailored for very specific sentences. The number rules seems to grow at a constant rate as the corpus grows. Not good, but we're stuck with it. (see Gaizauskas paper).

Largest number of rules for *S*, *NP*, and *VP*, but reduced ambiguity.

Strategy

The task in grammar writing is to choose the best elements for N and P. Consider constructing a grammar for named entities (noun phrases), or for a large corpus of sentences (1M+ words).

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

Instructions for homework

[see hw1 pdf on website]