Knowledge Representation

Models to represent Linguistic Knowledge

- We will use certain ***models to represent the required linguistic knowledge.***
- **State Machines** -- FSAs, FSTs, HMMs, ATNs, RTNs
- **Formal Rule Systems** -- Context Free Grammars, Unification Grammars, Probabilistic CFGs.
- Other common representational formalisms:
  - first order predicate logic
  - conceptual dependency graphs
  - semantic networks
  - Frame-based representations

Difficulty in NLP understanding

- arises from:
- Natural language is extremely **rich** in form and structure:
  - How to represent meaning,
  - Which structures map to which meaning structures.
- Natural language is **very ambiguous*.*** One input can mean many different things. Ambiguity can be at different levels.
  - Phonics Level: different meaning for the same sound
  - Lexical (word level) ambiguity -- different meanings of words
  - Syntactic ambiguity -- different ways to parse the sentence
  - Interpreting partial information -- how to interpret pronouns
  - Contextual information -- context of the sentence may affect the meaning of that sentence.

Speech Recognition - Complications

- No simple mapping between sounds and words
  - Variance in pronunciation due to gender, dialect, …
    - Restriction to handle just one speaker
  - Same sound corresponding to diff. words
    - e.g. bear, bare
  - Finding gaps between words
    - “how to recognize speech”
    - “how to wreck a nice beach”
  - Noise

Syntactic Analysis: Complications

- Rules of syntax (grammar) specify the possible organization of words in sentences and allows us to determine sentence’s structure(s)
  - “John saw Mary with a telescope”
    - John saw (Mary with a telescope)
    - John (saw Mary with a telescope)
  - “fruit flies like a banana”
- Parsing: given a sentence and a grammar
  - Checks that the sentence is correct according to the grammar and if so returns a **parse tree** representing the structure of the sentence

Semantic Analysis – Complications Ambiguous Example

- Some interpretations of : I made her duck

Semantic Analysis – Complications Ambiguous Example

- Some interpretations of : I made her duck
  1. I cooked ***duck*** for her.
  2. I cooked ***duck*** belonging to her.
  3. I created a toy duck which she owns.
  4. I caused her to quickly lower her head or body.
  5. I used magic and turned her into a ***duck***.
- duck – morphologically and syntactically ambiguous: noun or verb.
- her – syntactically ambiguous: for her/ to her/ her
- make – semantically ambiguous: cook or create.
- make – syntactically ambiguous:
- Transitive – takes a direct object. => 2
- Di-transitive – takes two objects. => 5
- Takes a direct object and a verb. => 4

The derivation of a sentence

➢One difficulty, that can add huge complexity to the parsing problem is: Determining which of several applicable rules should be used at any step of the derivation.

➢If a wrong choice is made, the parser may fail to recognize a legal sentence.

➢The problem of selecting the correct rule at any stage of the parse is handled either by

→allowing the parser to set backtrack pointers and return if an incorrect choice was made

→ ***using*** look-ahead to check the input string for features that will help determine the proper rule to apply.

Grammar types

- Regular: nonterminal ----> terminal[nonterminal]

  S ----> aS

- Context-free: nonterminal ----> anything

  S ----> aSb

- Context-sensitive: more nonterminals on right-hand side

  ASB ----> AAaBB

- Recursively enumerable: no constraints

  Natural languages probably is considered (dealt with as) Context sensitive

Syntactic Processing

- **Parsing --** converting a flat input sentence into a hierarchical structure that corresponds to the units of meaning in the sentence.

  • There are different parsing formalisms and algorithms.
  • Most formalisms have two main components:
  → **grammar --** a declarative representation describing the syntactic structure of sentences in the language.
  → **parser --** an algorithm that analyzes the input and outputs its structural representation (its parse) consistent with the grammar specification.
  CFGs are in the center of many of the parsing mechanisms. But they are complemented by some additional features that make the formalism more suitable to handle natural languages.

Context-free grammars

We will look at the simplest Context-Free Grammars,

```
sentence    → noun_phrase  verb_phrase

noun_phrase → proper_name
noun_phrase → article  noun

verb_phrase → verb
verb_phrase → verb  noun_phrase
verb_phrase → verb  noun_phrase
                     prep_phrase
verb_phrase → verb  prep_phrase

prep_phrase → preposition  noun_phrase
```

Context-free grammars

The still-undefined syntactic units are ***preterminals***. They correspond to parts of speech. We can define them by adding lexical productions to the grammar:

article → **the** | **a** | **an**

noun → **pizza** | **bus** | **boys** | ...

preposition → **to** | **on** | ...

proper_name → **Jim** | **Dan** | ...

verb → **ate** | **yawns** | ...

This is not practical on a large scale. Normally, we have a lexicon (dictionary) stored in a database, that can be interfaced with the grammar.

Context-free grammars

Derivation of a sentence:

```
Sentence →
noun_phrase verb_phrase →
proper_name verb_phrase →
Jim verb_phrase →
Jim verb noun_phrase prep_phrase →
Jim ate noun_phrase prep_phrase →
Jim ate article noun prep_phrase →
Jim ate a noun prep_phrase →
Jim ate a pizza prep_phrase →
Jim ate a pizza preposition noun_phrase →
Jim ate a pizza on noun_phrase →
Jim ate a pizza on article noun →
Jim ate a pizza on the noun →
Jim ate a pizza on the bus
```

Direction of parsing

In practice, parsing is never "pure".

- ➢Top-down, enriched: check data early to discard wrong hypotheses (somewhat like recursive-descent parsing in compiler construction).
- ➢Bottom-up, enriched: use productions, suggested by data, to limit choices
- ➢A popular bottom-up analysis method: chart parsing.
- ➢Popular top-down analysis methods:
  - transition networks (used with Lisp),
  - logic grammars (used with Prolog).

Top Down Example.

Consider the following context free grammar

1. Sentence→ Noun_phrase Verb_phrase
2. Noun_phrase → Noun
3. Noun_phrase → Article Noun
4. Verb_phrase →Verb
5. Verb_phrase →Verb Noun_phrase
6. Article → a
7. Article → the
8. Noun → man
9. Noun → dog
10. Verb → likes
11. Verb → bites

The derivation of a sentence

A Top-Down derivation:

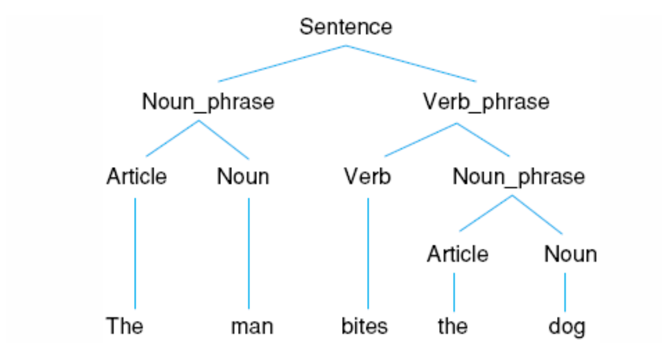
String	Rule
Sentence	1
Noun_phrase Verb_phrase	3
Article Noun Verb_phrase	7
the Noun Verb_phrase	8
the man Verb_phrase	5
the man Verb Noun_phrase	11
the man bites Noun_phrase	3
the man bites Article Noun	7
the man bites the Noun	9
the man bites the dog	

1. Sentence→ Noun_phrase Verb_phrase
2. Noun_phrase → Noun
3. Noun_phrase → Article Noun
4. Verb_phrase →Verb
5. Verb_phrase →Verb Noun_phrase
6. Article → a
7. Article → the
8. Noun → man
9. Noun → dog
10. Verb → likes
11. Verb → bites

Do it Bottom-Up

The parse Tree

Parse tree for the sentence “The man bites the dog.”



Syntactic Analysis: Complications

John saw Mary in a park with a telescope.