# Theory of Computing

Sujata Dongre

# The Standard Model

- Statistical methods are used for understanding NLP and NLU
- NLU involves tasks like parsing sentences, assigning semantic relations to parts of sentences

# The Standard Model (cont..)

- Morphology, syntax, semantics and pragmatics are the four parts of any language
- Morphology: study of the structure of individual words. e.g "going" is the word "go"
- Lexicon: structure that keeps tracks of the words
- Part-of-speech: set of words with similar syntactic properties
- Words can have two types of features: number (singular, plural) and person(I, we)

# The Standard Model (cont..)

- After studying morphology, it is important to study syntactic structure of as it helps in determining meaning of the sentence
- Syntactic structure is built up with grammar i.e Context-Free Grammar
- It is possible to have two parse-trees for the same sentence using context-free grammars, in that case grammar of that language is said to be ambiguous

# The Standard Model (cont..)

- Chomsky-normal form: a regular context-free grammar except NT -> terminal or NT -> NT NT
- Limitations of CFGs:
  - English language subject-verb agreement. The solution is using multiplying out features
  - Long-distance dependencies. The solution is use slash categories. e.g s -> wh s/np which means that when a wh-word starts the sentence the rest of the sentence is missing an np at some point

# The Standard Model (cont..)

- Context-free parser: algorithm for finding the structure of the sentence as per the grammar
- Chart parsing: one of the parsing technique

# The Standard Model (cont..)

- Working of Chart Parser:
  - Data structures used are key lists(stack for words in a sentence), chart(consists of terminals and NTs), edges(shows the rule that can be applied)
  - Get the entry from key lists, if entry does not exist, add entry to the chart, remove entry from key list, add an edge for the rule being applied

# Chart Parser

- Let us construct a chart for the sentence "The biscuits ate dog salespeople."
- Grammar rules are:

  s-maj -> s fpunc

  s -> np vp

  vp -> verb

  vp -> verb noun noun

  np -> det noun

  np -> noun

# Chart Parser (cont..)

s-maj -> s fpunc

s -> np vp

np -> det noun

np -> noun

vp -> verb np np

det -> the

noun -> biscuits

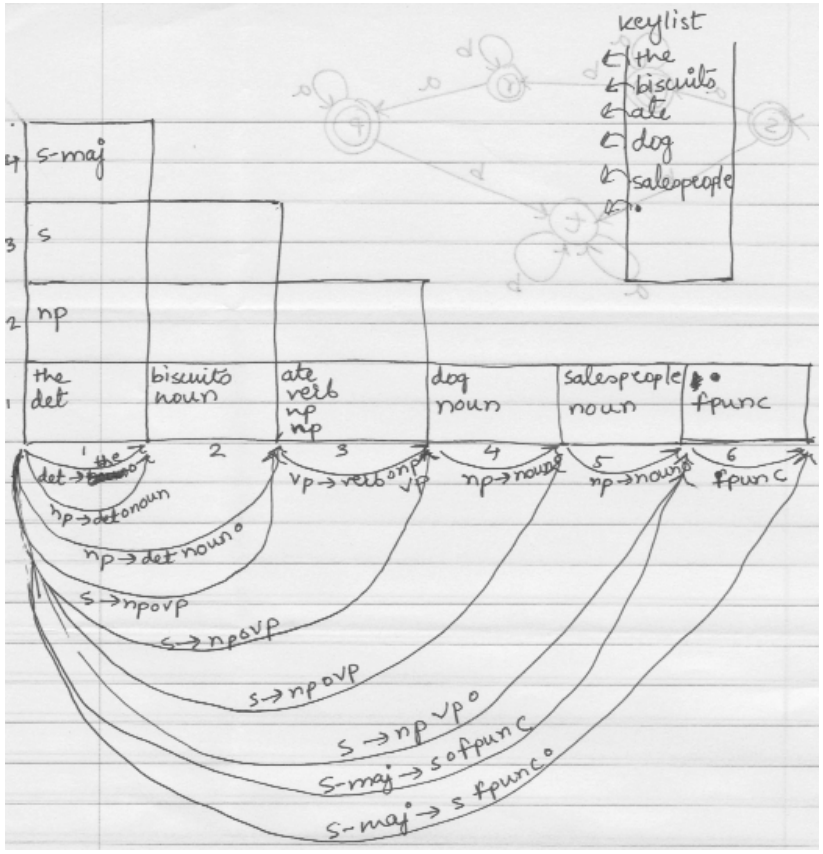
noun -> dog

noun -> salespeople

verb -> ate

fpunc -> .

# Chart Parser (cont..)



# The Standard Model (cont..)

- All words, sentences have meaning
- Meanings are senses of the words
- Understanding the text is understanding the intended sense for that particular instance

# The Standard Model (cont..)

- Compositional semantics: sentences get their meanings by combining the meanings of the individual words
- Selectional restrictions: particular senses place limits on how they can combine with other senses

# Deterministic Finite Automata

- Complexity of computer program = running time + auxiliary memory
- Running time: number of instructions being executed
- Auxiliary memory: main memory e.g. RAM

# DFA(cont..)

- is a finite automata model with unique output state depending on input and current state
- denoted by a quintuple $A = (Q, \Sigma, \delta, s, F)$

  $Q$ = finite set of states

  $\Sigma$ = finite input alphabet

  $\delta$ = transition function

  $s$ = initial state, $s \in Q$

  $F$ = set of favorable states, $F \subseteq Q$

# DFA (cont..)

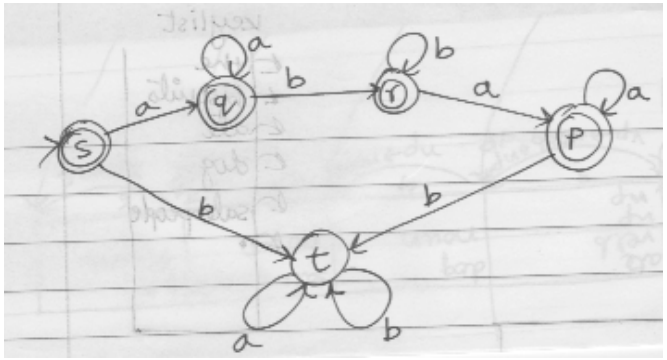
- Accepted state = after reading all the input alphabets, automata is in the favorable state
- Language of A = all input words accepted by A

# DFA (cont..)

- Examples:-

  DFA has some inputs, depending upon the input and the current state of the model, the next state is determined. By determinism, it means that the model has one and only one next output state for the combination of input and current state

# DFA (cont..)




- In this DFA, suppose we get input word as ‘aaabba’, we have to check whether the given word is accepted or rejected
- Only s, q, r and p are said to be accepted states while t is the rejected state

# DFA (cont..)

- Thus, by taking each input from word and the current state, we can determine if the string is accepted or not
- (s, a) = q, (q, a) = q, (q, a) = q,

  (q, b) = r, (r, b) = r, (r, a) = p
- As the p is the one of the accepted states for this automata, we can say that the string ‘aaabba’ is accepted

# DFA (cont..)

- The language accepted by automata should be null string/strings starts with a/no more than one occurrence of string aab in the input
- Hence the string ‘aabbaab’ is rejected by the automata as it has two occurrences of ‘aab’

# Nondeterministic Finite Automata

- is a finite automata model with several next states for given input and current state
- denoted by a quintuple $A = (Q, \Sigma, \Delta, s, F)$

  $Q$ = finite set of states

  $\Sigma$ = finite input alphabet

  $\Delta$ = transition relation

  $s$ = initial state, $s \in Q$

  $F$ = set of favorable states, $F \subseteq Q$

# NFA (cont..)

- NFA are easier to design than DFA
- Computational power of DFA and NFA is the same

# Determinism Vs. Nondeterminism

- NFA can be converted to equivalent DFA
- Conversion from NFA to DFA may cause increase in number of states exponentially
- For pattern matching problem, the equivalent DFA has the same number of states as NFA

# Regular Expressions

- provides means for identifying words, patterns of characters
- Set-theoretical operations: union, intersection, complementation, concatenation and Kleene star
- Regular language: language described by a regular expression

# Regular Expression (cont..)

- Similarity between regular expression and search query
- Search queries uses wild cards for searching on web pages
- Use Kleene star for the search words and we will get the same match as the above search query
- Any page that doesn't match the query will not be accepted by finite automata as well
- UNIX 'grep' command: returns all lines that match the pattern

# Context Free Grammar

- Consists of a set of rules with the derivation mechanism
- Context-free means that the nonterminals that appear in rules can be changed regardless of context in which they occur

# Context Free Grammar (cont..)

- nonterminals: characters that never appear in the output string
- terminals: will appear in the output string of the derivation process
- Context-free grammar G is a quadruple(Σ, NT, RS, S)

  Σ = set of inputs(terminals)

  NT = set of nonterminals

  R = set of rules

  S = starting symbol, $S \subseteq NT$

# Context Free Grammar (cont..)

- derivation is a nondeterministic process of applying different rules to occurrences of nonterminals at any step
- L(G) is a context-free language if grammar G is context-free

# CFG (cont..)

- Each language has some specific grammar structure to follow. CFG also consists of a set of rules that can be used to derive a string
- Examples:-

  Consider the grammar

  G = S -> SS, S -> aS, S->b
- Let us derive the string ‘aabaab’ from the above set of rules

# CFG (cont..)

- S -> aS

  S -> aSS (using S -> SS)

  S -> aaSS (using S -> aS)

  S -> aabS (using S -> b)

  S -> aabaS (using S -> aS)

  S -> aabaaS (using S -> aS)

  S -> aabaaaS (using S -> aS)

  S -> aabaaab (using S -> b)

# CFG (cont..)

- The language L(G) for the above grammar is strings starting with a or b but it has to end with b
- L(G) = $\{a^n b^m \mid n = 0,1,2,\ldots \text{ and } m = 1,2,\ldots\}$

# Parsing

- determining the grammatical structure of the sentence with respect to a given formal grammar
- describes the structure of the statement, its meaning
- parse tree is formed by such derivation process in which nodes are the terminals and leaves are the terminals or e(empty string)
- two types of derivations leftmost and rightmost. in leftmost derivation, replace the leftmost nonterminal while in the rightmost derivation, replace the rightmost nonterminal

# Parsing (cont..)

- all derivations with same parse trees are equivalent
- if two or more distinct parse trees, then grammar G is ambiguous
- if ambiguous grammar => two different meanings of the same statement
- inherently ambiguous => context-free languages that have no unambiguous generating context-free grammars