USN | | | | | | | | | |

Internal Assessment Test 2 – July 2024

Sub:	Natural Language Processing					Sub Code:	21AI643	Branch:	AI&DS
Date:	08/07/2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	VI		OBE

Answer any FIVE FULL Questions — MARKS | CO | RBT

1 Solve to find the probability of test sentence S2 in the following training set [10] CO1 L3

S1: The Arabian Knights
S2: These are the fairy tales of the east
S3: The stories of the Arabian knights are translated in many languages

Bi-gram model:

P(the/<s>)=0.67 P(Arabian/the)=0.4 P(knights/Arabian)=1.0

P(are/these)=1.0 P(the/are)=0.5 P(fairy/the)=0.2 P(tales/fairy)=1.0
P(of/tales)=1.0 P(the/of)=1.0 P(east/the)=0.2

P(stories/the)=0.2 P(of/stories)=1.0 P(are/knights)=1.0
P(translated/are)=0.5 P(in/translated)=1.0 P(many/in)=1.0
P(languages/many)=1.0

Test sentence(s): These are the fairy tales of the east

P(The/<s>) x P(Arabian/the) x P(Knights/Arabian) x P(are/knights) x P(the/are) x P(fairy/the) x P(tales/fairy) x P(of/tales) x P(the/of) x P(east/the)

= 0.67 x 0.4 x 1.0 x 1.0 x 0.5 x 0.2 x 1.0 x 1.0 x 1.0 x 0.2

=0.0067

2 What are the advantages and disadvantages of top-down and bottom-up parsing and give top-dowm and bottom-up search space for the sentence, 'paint the door', by applying the following grammar - [2+2+3+3] CO2 L3

S -> NP VP
S -> VP
NP -> Det Nominal
NP -> Noun
NP -> Det Noun PP
VP -> Verb NP
VP -> Verb
PP -> Preposition NP
Det -> this | that | a | the
Nominal -> Noun
Verb -> sleeps | paint | open | sings
Preposition -> from | with | on | to
Pronoun -> She | he | they
Nominal -> Noun Nominal

- **Left recursion**, which causes search to get stuck in an infinite loop.
- **Structural Ambiguity**, occurs when a word has more than one part-of-speech associated with it.
- **Attachment ambiguity**, if a constituent fits more than one position in a parse tree.
- **Coordination ambiguity,** occurs when it is not clear which phrases are being combined with a conjunction like and.
- **Local ambiguity,** occurs when certain parts of a sentence are ambiguous.
- Another problem with basic top-down strategy is that of **repeated parsing**.

S → NP VP	VP → Verb NP
S → VP	VP → Verb
NP → Det Nominal	PP → Preposition NP
NP → Noun	Det → this | that | a | the
NP → Det Noun PP	Verb → sleeps | paint | open | sings
Nominal → Noun	Preposition → from | with | on | to
Nominal → Noun Nominal	Pronoun → She | he | they

Paint the door.

A top-down search begins with the start symbol of the gramma the first level (ply) search tree consists of a single node labelled grammar in Table 4.2 has two rules with S on their left hand sid

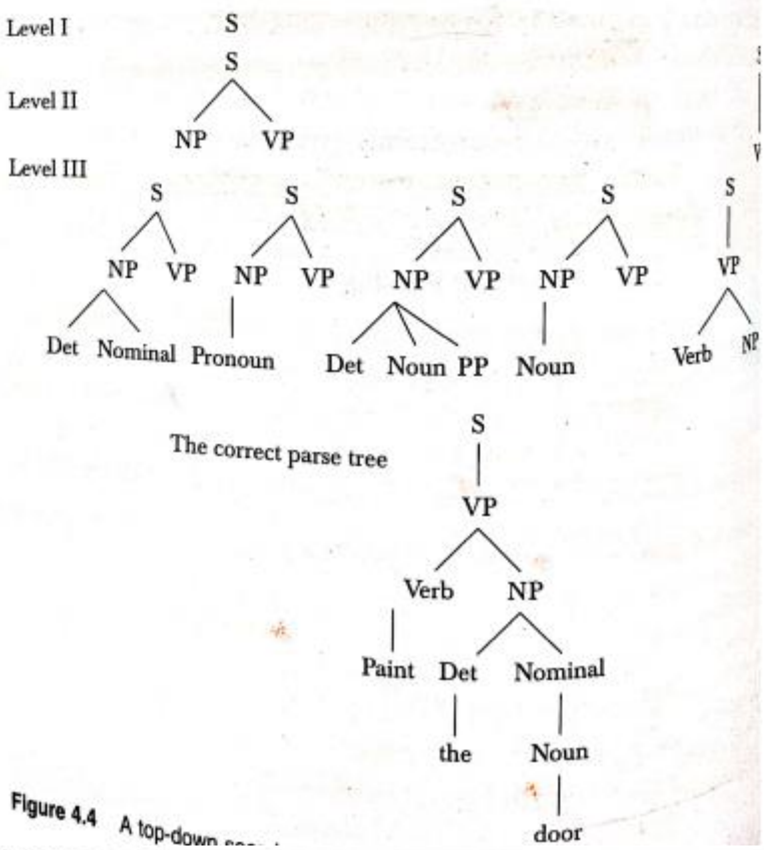


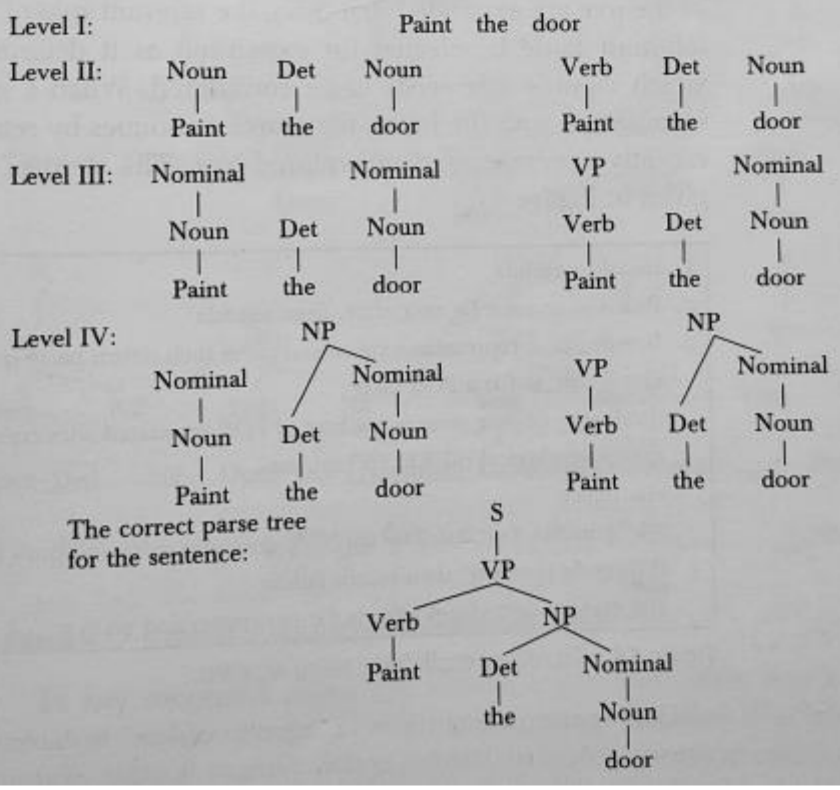
Figure 4.4 A top-down sea...

rules are used to expand the tree, which gives us two partial trees at the second level search, as shown in Figure 4.4. The third level is generated by expanding the non-terminal at the bottom of the search tree in the previous ply. Due to space constraints, only the expansion corresponding to the left-most non-terminals has been shown in the figure. The subsequent steps in the parse are left, as an exercise, to the readers. The correct parse tree shown in Figure 4.4 is obtained by expanding the fifth parse tree of the third level.

<table>
<tr><td></td><td>

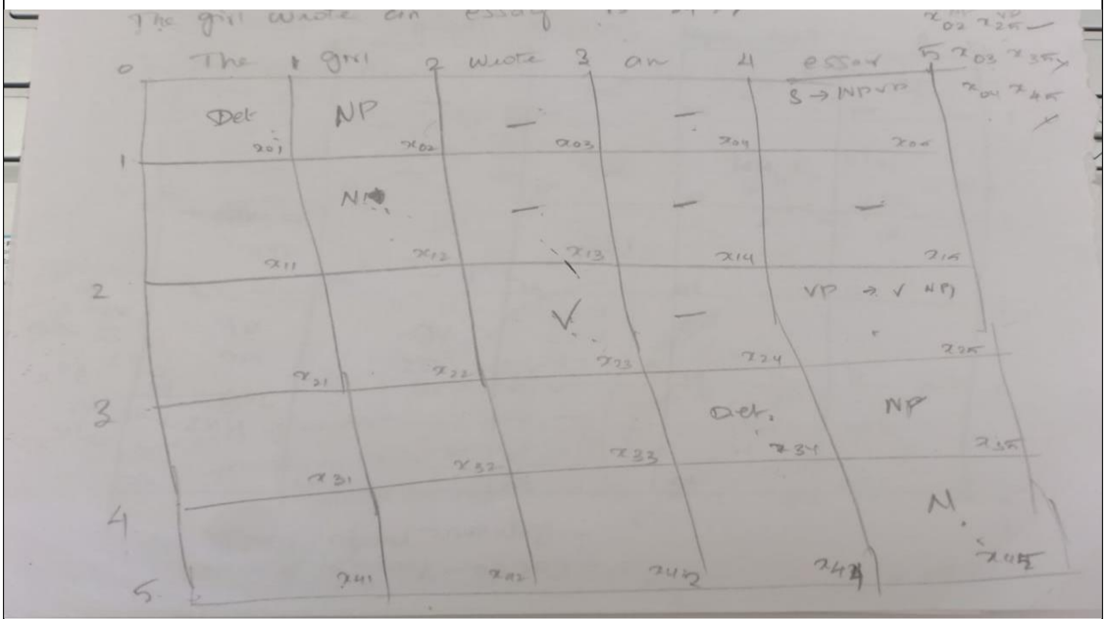
4.4.2 Bottom-up Parsing

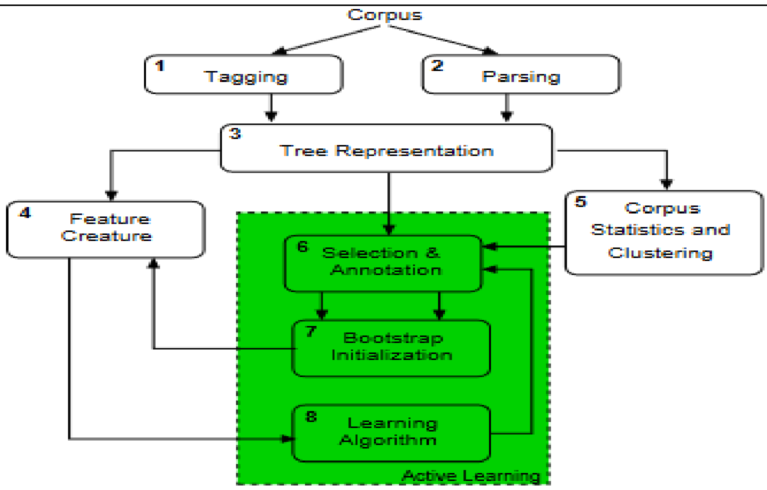
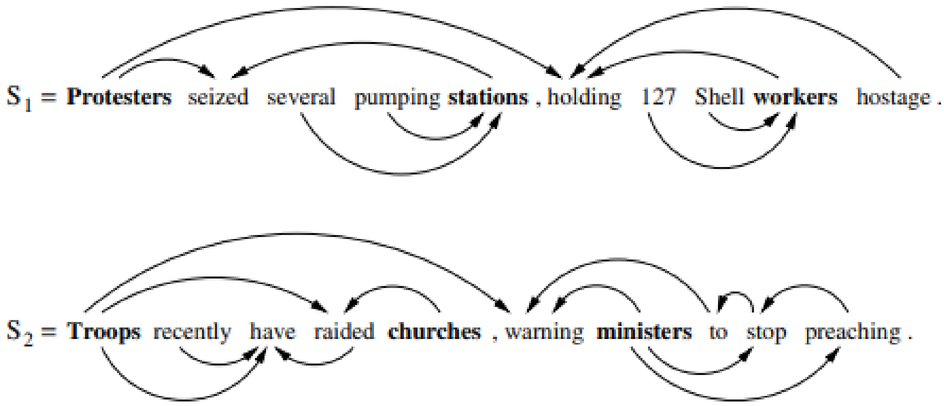
A bottom-up parser starts with the words in the input sentence and attempts to construct a parse tree in an upward direction towards the root. At each step, the parser looks for rules in the grammar where the right hand side matches some of the portions in the parse tree constructed so far, and reduces it using the left hand side of the production. The parse is considered successful if the parser reduces the tree to the start symbol of the grammar. Figure 4.5 shows some steps carried out by the bottom-up parser for sentence (4.7).

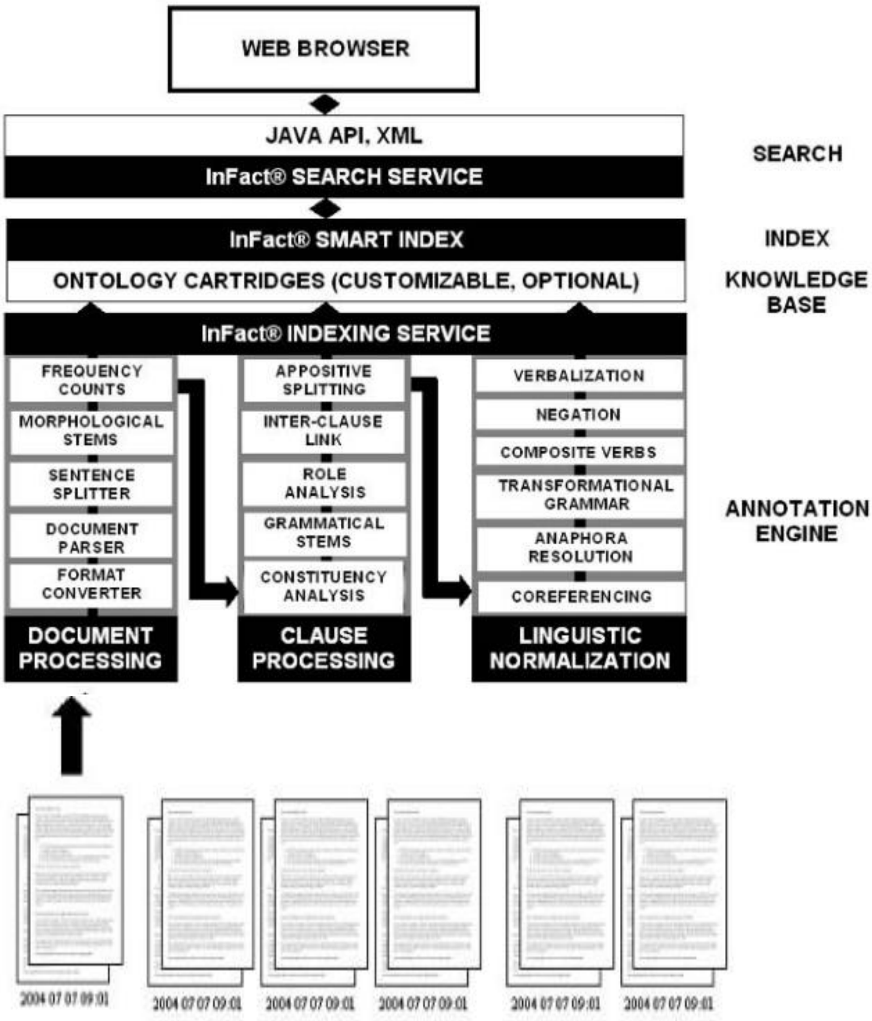


</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>Design CYK algorithm. Tabulate the sequence of states created by CYK algorithm while parsing the sentence, “A pilot like flying planes”. Consider the following simplified grammar in CNF
S -> NP VP   NN -> Pilot   VBG -> flying
NP -> DT NN   NNS -> plane   JJ -> flying
NP -> JJ NNS   VP -> VBG NNS   Det -> a
VP -> VBZ NP   VBZ -> likes</td><td>[5+5]</td><td>CO2</td><td>L3</td></tr>
</table>

<table>
<tr><td></td><td>Let $w = w_1\ w_2\ w_3\ w_i \ldots w_j \ldots w_n$
and $w_{ij} = w_i \ldots w_{i+j-1}$
// Initialization step
for $i := 1$ to n do
for all rules $A \rightarrow w_i$ do
chart $[i,1] = \{A\}$
// Recursive step
for $j = 2$ to n do
for $i = 1$ to $n-j+1$ do
begin
chart $[i, j] = \phi$
for $k = 1$ to $j-1$ do
chart $[i, j] :=$ chart$[i, j] \cup \{A \mid A \rightarrow BC$ is a production and
$B \in$ chart$[i, k]$ and $C \in$ chart $[i+k, j-k]\}$
end
if $S \in$ chart$[1, n]$ then accept else reject
Figure 4.12 The CYK algorithm


</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>With a neat diagram, explain the architecture used in the task of learning to annotate cases with knowledge roles.</td><td>[10]</td><td>CO3</td><td>L2</td></tr>
</table>

	 **Fig. 1.5.** The Learning Framework Architecture.			
5	Explain Dependency Path Kernel for relation extraction. The pattern examples show the two entity mentions, together with the set of words that are relevant for their relationship. A closer analysis of these examples reveals that all relevant words form a shortest path between the two entities in a graph structure where edges correspond to relations between a word (head) and its dependents. For example, Figure 3.4 shows the full dependency graphs for two sentences from the ACE (Automated Content Extraction) newspaper corpus, in which words are represented as nodes and word-word dependencies are represented as directed edges. A subset of these word-word dependencies captures the predicate-argument relations present in the sentence. Arguments are connected to their target predicates either directly through an arc pointing to the predicate ('troops → raided'), or indirectly through a preposition or infinitive particle ('warning ← to ← stop'). Other types of word-word dependencies account for modifier-head relationships present in adjective-noun compounds ('several → stations'), noun-noun compounds ('pumping → stations'), or adverb-verb constructions ('recently → raided'). Word-word dependencies are typically categorized in two classes as follows: • [Local Dependencies] These correspond to local predicate-argument (or head modifier) constructions such as 'troops → raided', or 'pumping → stations' in Figure 3.4. • [Non-local Dependencies] Long-distance dependencies arise due to various linguistic constructions such as coordination, extraction, raising and control. In Figure 3.4, among non-local dependencies are 'troops → warning', or 'ministers → preaching'. A Context Free Grammar (CFG) parser can be used to extract local dependencies, which for each sentence form a dependency tree. Mildly context sensitive formalisms such as Combinatory Categorial Grammar (CCG) model word-word dependencies more directly and can be used to extract both local and long-distance dependencies, giving rise to a directed acyclic graph,  **Fig. 3.4.** Sentences as dependency graphs.	[10]	CO3	L2
6	Explain Functional overview of InFact system.	[10]	CO3	L2


WEB BROWSER
JAVA API, XML
InFact® SEARCH SERVICE
SEARCH
InFact® SMART INDEX
INDEX
ONTOLOGY CARTRIDGES (CUSTOMIZABLE, OPTIONAL)
KNOWLEDGE BASE
InFact® INDEXING SERVICE
FREQUENCY COUNTS
MORPHOLOGICAL STEMS
SENTENCE SPLITTER
DOCUMENT PARSER
FORMAT CONVERTER
APPOSITIVE SPLITTING
INTER-CLAUSE LINK
ROLE ANALYSIS
GRAMMATICAL STEMS
CONSTITUENCY ANALYSIS
VERBALIZATION
NEGATION
COMPOSITE VERBS
TRANSFORMATIONAL GRAMMAR
ANAPHORA RESOLUTION
COREFERENCING
ANNOTATION ENGINE
DOCUMENT PROCESSING
CLAUSE PROCESSING
LINGUISTIC NORMALIZATION
2004 07 07 09:01
2004 07 07 09:01
2004 07 07 09:01
2004 07 07 09:01
2004 07 07 09:01
2004 07 07 09:01