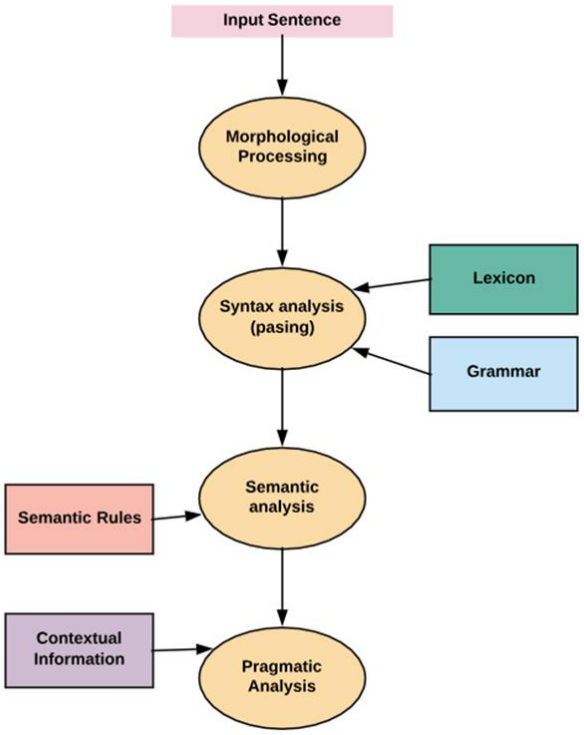
Sr. No.		Question	Marks	CO Bloom's Level
Q.1	[A]	Explain the various stages in Natural Language Processing.	05	CO1 (L2)

Solution:



- Morphological and Lexical Analysis
- Syntactic Analysis
- Semantic Analysis
- Discourse Integration
- Pragmatic Analysis

Morphological and Lexical Analysis

Lexical analysis is a vocabulary that includes its words and expressions. It depicts analyzing, identifying and description of the structure of words. It includes dividing a text into paragraphs, words and the sentences

Individual words are analyzed into their components, and nonword tokens such as punctuations are separated from the words.

Semantic Analysis

Semantic Analysis is a structure created by the syntactic analyzer which assigns meanings. This component transfers linear sequences of words into structures. It shows how the words are associated with each other.

Semantics focuses only on the literal meaning of words, phrases, and sentences. This only abstracts the dictionary meaning or the real meaning from the given context. The structures assigned by the syntactic analyzer always have assigned meaning

E.g.. "colorless green idea." This would be rejected by the Symantec analysis as colorless Here; green doesn't make any sense.

Pragmatic Analysis

Pragmatic Analysis deals with the overall communicative and social content and its effect on interpretation. It means abstracting or deriving the meaningful use of language in situations. In this analysis, the main focus always on what was said in reinterpreted on what is meant.

Pragmatic analysis helps users to discover this intended effect by applying a set of rules that characterize cooperative dialogues.

E.g., "close the window?" should be interpreted as a request instead of an order.

Syntax analysis

The words are commonly accepted as being the smallest units of syntax. The syntax refers to the principles and

rules that govern the sentence structure of any individual languages.
Syntax focus about the proper ordering of words which can affect its meaning. This involves analysis of the words in a sentence by following the grammatical structure of the sentence. The words are transformed into the structure to show hows the word are related to each other.

Discourse Integration

It means a sense of the context. The meaning of any single sentence which depends upon that sentences. It also considers the meaning of the following sentence.
For example, the word "that" in the sentence "He wanted that" depends upon the prior discourse context.

Q.1	[B]	Identify the type of ambiguity in the following sentences: i. I went to the bank. ii. Flying planes can be dangerous. iii. Visiting relatives can be boring. iv. The burglar threatened the student with the knife. v. I invited the person with the microphone.	05	CO1 (L3)

Solution: (type of ambiguity with explanation is expected)

i. I went to the bank.- Lexical ambiguity
ii. Flying planes can be dangerous. - Syntactic ambiguity
iii. Visiting relatives can be boring.- Syntactic ambiguity
iv. The burglar threatened the student with the knife.- Syntactic ambiguity
v. I invited the person with the microphone.- Syntactic ambiguity

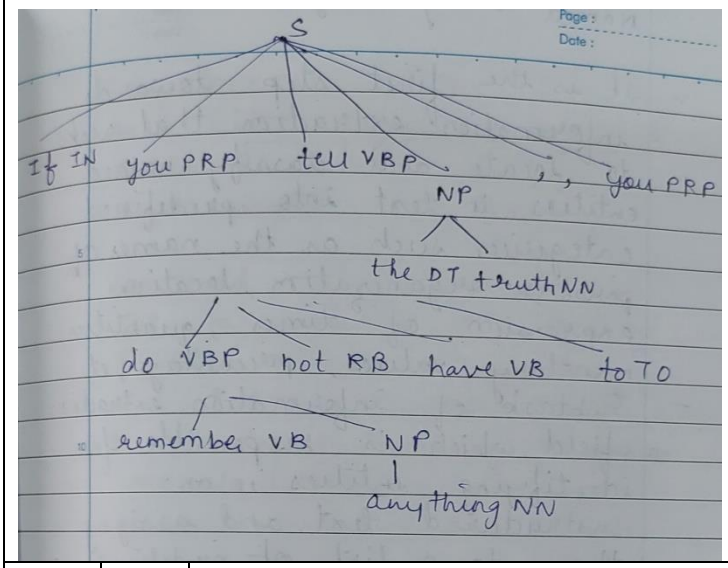
Q.2	[A]	Apply shallow parsing (chunking) to the following input sentence. Also draw the parse tree structure. Input: If you tell the truth, you don't have to remember anything.	05	CO3 (L3)

Solution:

Tokens: ['If', ' you', 'tell', 'the', 'truth', ',', ' you', 'don't', 'have', 'to', 'remember', 'anything', '.']
POS Tagging: [('If', 'IN)', (' you', 'PRP'), ('tell', 'VB'), ('the', 'DET'), ('truth', 'NN'), (',', ','), (' you', 'PRP'), ('do', 'VB'), ('not', 'RB'), ('have', 'VB'), ('to', 'TO'), ('remember', 'VB'), ('anything', 'NN'), ('.', '.')]
Grammar declaration: NP: {<DT>?<JJ>*<NN>}
Chunking output:

```
(S
(If/IN
You/PRP
Tell/VB
(NP the/DT truth/VB
,/,
you/PRP
do/VB
not/RB
have/VB
to/TO
remember/VB
(NP anything/NN)
./.
)
```

Parse tree:



Q.2	[B]	Generate the regular expression for: i. 'a' followed by zero or more 'b', ii. 'a' followed by 5 'b's', iii. Sequence of lower case letters joined with an underscore, iv. Sequence of one upper case letter followed by lower case letter. v. A string that has an 'a' followed by anything ending in 'b'.	05	CO3 (L3)

Solution:

i. 'a' followed by zero or more 'b',

^a(b*)$

ii. 'a' followed by 5 'b's',

ab{5}?

iii. Sequence of lower case letters joined with an underscore,

^[a-z]+_[a-z]+$

iv. Sequence of one upper case letter followed by lower case letter.

[A-Z]+[a-z]+$

v. A string that has an 'a' followed by anything ending in 'b'.

a.*?b$

Q.3	[A]	Compare top-down and bottom-up parsing approaches.	05	CO3 (L3)

Solution:

Top-down Parsing	Bottom-up Parsing
It is a parsing strategy that first looks at the highest level of the parse tree and works down the parse tree by using the rules of grammar.	It is a parsing strategy that first looks at the lowest level of the parse tree and works up the parse tree by using the rules of grammar.
Top-down parsing attempts to find the left most derivations for an input string.	Bottom-up parsing can be defined as an attempt to reduce the input string to start symbol of a grammar.
In this parsing technique we start parsing from top (start symbol of parse tree) to down (the leaf node of parse tree) in top-down manner.	In this parsing technique we start parsing from bottom (leaf node of parse tree) to up (the start symbol of parse tree) in bottom-up manner.
This parsing technique uses Left Most Derivation.	This parsing technique uses Right Most Derivation.

Its main decision is to select what production rule to use in order to construct the string.	Its main decision is to select when to use a production rule to reduce the string to get the starting symbol.
A top-down parser can be easily structured and formed.	It is difficult to produce a bottom-up parser
It is less powerful.	It is more powerful.
Error detection is easy.	Error detection is difficult.

Q.3	[B]	Apply the CKY parsing algorithm to the generate the parse tree for the given input string: Input Sentence: The flight includes a meal. Grammar: S→NP VP NP→ DET N VP→ V NP V→includes DET→ the DET→a N→meal N→flight	05	CO3 (L3)

Solution:

CYK algorithm is a parsing algorithm for context free grammar.
In order to apply CYK algorithm to a grammar, it must be in Chomsky Normal Form. It uses a dynamic programming algorithm to tell whether a string is in the language of a grammar.

1. Let n be the number of words in the input. Think about n+1 lines separating them, numbered 0 to n.
2. Xij will denote the words between line i and j.
3. We build a table so that Xij contains all the possible non-terminal spanning for words between line I and j.
4. We build the table bottom-up.