P.V.P Siddhartha Institute of Technology (Autonomous)

Department of Computer Science and Engineering

NATURAL LANGUAGE PROCESSING III CSE SEM-II

Descriptive-II Answers Academic Year: 2025-26

Subject Code: 23CS4602C | Regulation: PVP23

1(a). Apply treebank data to analyze the syntactic structure of given sentences. (CO2-L3) (5M)

Ans:

Definition

A treebank is a corpus of sentences annotated with their syntactic structure. A constituency treebank groups words into phrases such as noun phrases (NPs) and verb phrases (VPs); a dependency treebank records head-dependent relations between words.

Step 1: Select a sentence and a suitable treebank

Consider **"The student reads a book."** Use Penn Treebank-style phrase labels and part-of-speech tags. The question supplies no particular sentence, so this sentence is used as a worked example.

Word	The	student	reads	a	book
POS tag	DT	NN	VBZ	DT	NN

Step 2: Identify constituents

The student is the subject NP. **a book** is the object NP. **reads a book** is the VP. The complete sentence consists of a subject NP followed by a VP.

(S (NP (DT The) (NN student)) (VP (VBZ reads) (NP (DT a) (NN book))))

Step 3: Derive grammar rules from the annotation

S → NP VP; NP → DT NN; VP → VBZ NP

Lexical rules associate DT with "The" and "a", NN with "student" and "book", and VBZ with "reads". Applying these rules reconstructs the annotated tree.

Step 4: Analyze a second sentence

For **"The students read books"**, the subject NP has DT + NNS, the VP has VBP + NP, and the object NP contains NNS. Comparing the two trees shows that phrase structure remains similar while number and verb-form tags change.

Result and use

Treebank annotations reveal phrase boundaries, grammatical organization, and alternative structures. A parser learns structural patterns from annotated examples and predicts trees for new sentences. Treebanks support parser training, evaluation, and syntactic ambiguity analysis; a particular treebank annotation represents a chosen analysis.

1(b). Utilize dependency parsing techniques to represent sentence structure. (CO2-L3) (5M)

Ans:

Definition

Dependency parsing represents a sentence as directed, labeled relations between words. Each relation connects a governing word, called the **head**, to a **dependent**. In a basic dependency tree, every word except the root has exactly one head.

Step 1: Tokenize the sentence

Example: **“The student reads a book.”** Tokens are The, student, reads, a, book, and the full stop. The following analysis uses Universal Dependencies-style relation labels.

Step 2: Choose the root

The main verb **reads** is the root because it expresses the central predicate. “student” identifies who reads, and “book” identifies what is read.

Step 3: Assign heads and relation labels

ID	Token	Head	Relation	Interpretation
1	The	student	det	Determiner of subject
2	student	reads	nsubj	Nominal subject
3	reads	ROOT	root	Main predicate
4	a	book	det	Determiner of object
5	book	reads	obj	Direct object
6	.	reads	punct	Sentence punctuation

Step 4: Read the dependency representation

ROOT → reads; reads → student [nsubj]; reads → book [obj]

student → The [det]; book → a [det]; reads → . [punct]

The labels distinguish grammatical functions even when word order changes. The determiners depend on nouns rather than directly on the main verb.

Parsing techniques

Transition-based parsing builds a tree through actions such as SHIFT, LEFT-ARC, and RIGHT-ARC using a stack and buffer. **Graph-based parsing** scores candidate dependency arcs and selects a high-scoring valid tree. Neural models can predict the scores or transition actions.

Result

The tree captures who performs the action and what receives it. Dependency relations are useful for relation extraction, question answering, information extraction, and identifying predicate arguments.

2(a). Construct parse tables using the CYK algorithm for any sentence. (CO2-L3) (5M)

Ans:

Definition

The Cocke-Younger-Kasami (CYK) algorithm is a bottom-up dynamic programming algorithm for context-free grammars in Chomsky Normal Form (CNF). It stores the nonterminals that derive each contiguous span of a sentence.

Step 1: Choose a sentence and CNF grammar

Sentence: **“she eats fish”**. Use the following grammar; each production has either two nonterminals or one terminal on its right-hand side.

S → NP VP; VP → V NP; NP → she | fish; V → eats

Step 2: Define the table and fill length-one spans

Let T[i,j] contain all nonterminals that derive words i through j. Since NP derives “she”, V derives “eats”, and NP derives “fish”:

T[1,1] = {NP}; T[2,2] = {V}; T[3,3] = {NP}

Step 3: Fill longer spans using binary rules

For each split k, insert A into T[i,j] when A → B C, B is in T[i,k], and C is in T[k+1,j].

Span [1,2], "she eats": NP followed by V matches no binary production. Therefore T[1,2] is empty.
Span [2,3], "eats fish": V followed by NP matches VP → V NP. Therefore T[2,3] = {VP}.
Span [1,3], "she eats fish": split after "she" gives NP followed by VP. By S → NP VP, T[1,3] = {S}. The other split produces no additional entry.

Start / End	1: she	2: eats	3: fish
1: she	{NP}	Empty	{S}
2: eats	-	{V}	{VP}
3: fish	-	-	{NP}

Step 4: Acceptance and parse reconstruction

The sentence is accepted because the start symbol S occurs in T[1,3]. Store the successful rule and split as backpointers to reconstruct the parse.

(S (NP she) (VP (V eats) (NP fish)))

Complexity: Standard CYK takes $O(n^3 |P|)$ time and $O(n^2 |N|)$ membership storage, where P is the binary production set and N is the nonterminal set. For a fixed grammar, these are $O(n^3)$ and $O(n^2)$.

2(b). Apply feature structures to represent grammatical information such as number and gender. (CO2-L3) (5M)

Ans:

Definition

A feature structure represents linguistic information as attribute-value pairs. Attributes may include category, number, person, gender, tense, and case. An attribute-value matrix (AVM) gives a compact representation of these features.

Step 1: Represent a noun phrase

For the English pronoun **"she"**, a simplified feature structure is:

Attribute	Value
CATEGORY	NP
PERSON	3
NUMBER	singular
GENDER	feminine
CASE	nominative

A feature structure for "students" can contain CATEGORY = N, NUMBER = plural, and PERSON = 3. Gender may be left unspecified because the word itself does not determine it.

Step 2: State an agreement constraint

In a simple present-tense clause, the subject and finite verb must have compatible person and number features. The verb "reads" requires a third-person singular subject; "read" is compatible with a third-person plural subject.

S → NP VP, with NP.AGR = VP.AGR

she.AGR = [PERSON: 3, NUMBER: singular]

reads.AGR = [PERSON: 3, NUMBER: singular]

Step 3: Apply unification

Unification combines compatible feature information. In **"She reads"**, both agreement structures specify person 3 and singular number, so they unify successfully. In **"They reads"**, plural and singular conflict, so unification fails. The corrected sentence is **"They read."**

Step 4: Use gender information

In **"Ravi said that he was ready"**, an illustrative representation may assign masculine and singular features to Ravi and "he" when those features are known from context. Their compatibility supports the proposed reference link, although feature agreement alone cannot prove coreference.

Result: Feature structures allow a grammar to express agreement and other constraints without writing a separate phrase rule for every combination. English present-tense verbs agree in person and number, while gender is relevant mainly to pronoun interpretation rather than verb agreement.

3(a). Explain how First-Order Logic supports formal semantic representation of natural language sentences. (CO4-L4) (5M)

Ans:

The question paper has an awkwardly worded opening ("Classify the ..."). This answer addresses its intended topic: formal semantic representation using First-Order Logic.

Definition

First-Order Logic (FOL) is a formal language for representing individuals, their properties, and relations between individuals. It supports precise truth conditions through predicates, variables, constants, connectives, and quantifiers.

Step 1: Identify the components

Component	Role	Example
Constant	Names an individual	ravi
Variable	Ranges over individuals	x, y
Predicate	Property or relation	Student(x), Reads(x,y)
Quantifier	Universal or existential scope	∀x, ∃x
Connective	Combines propositions	¬, ∧, ∨, →

Step 2: Translate sentences into formulas

"Ravi is a student." Student(ravi)

"Every student reads a book." ∀x (Student(x) → ∃y (Book(y) ∧ Reads(x,y)))

The universal quantifier ranges over all individuals. The implication restricts the claim to students. Each student must read at least one book; different students may read different books.

"Some student reads every book." ∃x (Student(x) ∧ ∀y (Book(y) → Reads(x,y)))

Here one student must read every book. Interchanging universal and existential quantifiers changes the meaning, so scope must be represented explicitly.

Step 3: Represent negation and inference

"No student sleeps." ∀x (Student(x) → ¬Sleeps(x))

Student(ravi), ∀x (Student(x) → Studies(x)) ⊨ Studies(ravi)

The inference follows by instantiating the universal statement for Ravi and applying modus ponens.

Role and limitation

FOL separates meaning from surface wording, enables logical inference, and makes ambiguity explicit. However, ordinary FOL does not automatically resolve reference or encode uncertainty, tense, and modality; these need additional representations or conventions.

3(b). Classify relations between word senses such as synonymy, antonymy, and hypernymy. (CO4-L4) (5M)

Ans:

Definition

A word sense is a particular meaning of a word in a context. Lexical semantic relations connect these senses. Classification must use the intended sense, because a word can have several meanings.

1. Synonymy: similarity of meaning

Two senses are synonyms when they have the same or closely similar meanings in a relevant context. Examples are **begin/start** and **large/big**. In “The class will begin at nine”, “start” preserves the main meaning. Synonyms are not always interchangeable in every expression: “big sister” does not generally mean “large sister”.

2. Antonymy: opposition of meaning

Type	Meaning	Example
Gradable	Opposite ends of a scale; intermediate values exist	hot / cold
Complementary	Opposing categories under the intended interpretation	alive / dead
Converse / relational	The same relation from opposite roles	buy / sell; parent / child

“Warm” lies between hot and cold, demonstrating gradability. If A buys an item from B, B sells that item to A, demonstrating a converse relation.

3. Hypernymy and hyponymy: category inclusion

A **hypernym** is a more general category; a **hyponym** is a more specific member category. “Animal” is a hypernym of “dog”; “dog” is a hyponym of “animal”. “Dog” and “cat” are co-hyponyms under “animal”.

poodle IS-A dog; dog IS-A animal; therefore poodle IS-A animal

The relation is directional: every dog is an animal, but every animal is not necessarily a dog. A diagnostic is the sentence “An X is a kind of Y”.

Worked classification

Pair	Relation	Reason
begin - start	Synonymy	Similar action meaning
hot - cold	Gradable antonymy	Opposite temperature values
rose - flower	Hyponymy / hypernymy	A rose is a kind of flower

Application: These relations support query expansion, semantic search, word-sense disambiguation, and lexical resources such as WordNet. A part-whole relation such as “wheel-car” is meronymy, not an IS-A relation.

---

4. Analyze supervised WSD by examining training data, features, and classification methods. (CO4-L4) (10M)

Ans:

Definition

Word Sense Disambiguation (WSD) is the task of selecting the intended sense of an ambiguous word from its context. In **supervised WSD**, a model learns a mapping from contextual features to sense labels using manually or otherwise reliably sense-annotated training examples.

Step 1: Define the sense inventory and training data

Consider the target word **bank**. Let S1 mean a financial institution and S2 mean the land beside a river. Every training occurrence is labeled according to the intended sense.

Training sentence	Sense label	Indicative context
She deposited money in the bank.	S1: financial	deposited, money
The bank approved the loan.	S1: financial	approved, loan
They sat on the river bank.	S2: river edge	sat, river
The boat reached the bank.	S2: river edge	boat, reached

A practical dataset should contain many varied examples per sense, with consistent annotation. Split data into training, validation, and test sets before selecting features or tuning parameters. Avoid placing duplicates or closely related document passages across splits.

Step 2: Preprocess and extract features

Local lexical features: words in a fixed window around “bank”; nearby bigrams and collocations such as “river bank” or “bank account”.
Grammatical features: POS tags, dependency relations, and governing verbs such as “deposit” or “sit”.
Wider context: sentence or document words indicating finance, geography, travel, or other topics.
Representations: binary word indicators, TF-IDF vectors, or contextual embeddings for the target token.

$x = [money, loan, river, boat]$

For a simple binary illustration, “She deposited money in the bank” becomes x = [1,0,0,0], labeled S1. “The boat reached the river bank” becomes x = [0,0,1,1], labeled S2. These four features are illustrative; a real model needs broader evidence.

Step 3: Train a classifier

Naive Bayes: combines a sense prior with likelihoods for contextual features. **SVM:** learns a separating decision boundary over feature vectors. **Logistic regression:** estimates conditional probabilities for senses. **Neural classifiers:** use learned or contextual representations and a sense-prediction layer. The choice depends on annotation volume, vocabulary coverage, and computation.

Step 4: Worked Naive Bayes classification

Suppose a trained multinomial Naive Bayes model has the following illustrative probabilities. They represent assumed values for this example, not measured experimental results. Other vocabulary items account for the remaining likelihood mass.

Quantity	Financial sense S1	River-edge sense S2
P(sense)	0.50	0.50
P(money | sense)	0.30	0.02
P(loan | sense)	0.20	0.01

For a new context containing one occurrence each of “money” and “loan”, score each sense by multiplying its prior and the two word likelihoods. The model assumes conditional independence of context words given the sense.

$$\text{Score}(s) = P(s) \times \prod P(w_i \mid s)$$

$$\text{Score}(S1) = 0.50 \times 0.30 \times 0.20 = 0.0300$$

$$\text{Score}(S2) = 0.50 \times 0.02 \times 0.01 = 0.0001$$

$$P(S1 \mid \text{context}) = 0.0300 / (0.0300 + 0.0001) \approx 0.9967$$

Therefore the model selects **S1: financial institution**. In practice, likelihoods require smoothing to avoid zero scores for unseen words, and log probabilities are used to prevent numerical underflow.

Step 5: Test and evaluate

$$\text{Accuracy} = \text{correctly predicted target senses} / \text{total test targets}$$

If 85 of 100 held-out occurrences are labeled correctly, accuracy is 85%. This is only an evaluation illustration. Report per-sense precision, recall, and F1 or macro-F1 when the sense distribution is imbalanced. Compare against a **most-frequent-sense baseline**, selected using training data alone.

Analysis of training data, features, and methods

Aspect	Effect on WSD performance
Training data	Rare senses need sufficient examples; inconsistent labels limit learning.
Features	Nearby collocations are strong clues; wider context can resolve less local ambiguity.
Classification method	Linear models suit sparse features; contextual encoders model richer context but require more resources.
Domain shift	A finance-trained model may overpredict the financial sense in geography text.

Advantages and limitations

Supervised WSD can be accurate when labeled examples cover the relevant senses and domain. Its main limitations are annotation cost, class imbalance, fixed sense inventories, unseen senses, and domain dependence. Strong results require reliable labels and evaluation on genuinely held-out contexts.

5(a). Develop a discourse representation using Rhetorical Structure Theory (RST). (CO3-L3) (5M)

Ans:

Definition

Rhetorical Structure Theory (RST) represents how spans of text are related to form coherent discourse. It links elementary discourse units (EDUs) through rhetorical relations such as Cause, Evidence, Elaboration, Contrast, and Condition.

Key concepts

EDU: a minimal discourse unit, commonly a clause, according to the chosen segmentation convention.
Nucleus: the central information in a nucleus-satellite relation.
Satellite: supporting information that explains, justifies, or elaborates the nucleus.
Multinuclear relation: a relation connecting spans with comparable importance, such as Contrast in many RST schemes.

Step 1: Segment the example

Text: **“Because it rained heavily, the match was cancelled. It will be played on Sunday instead.”**

EDU	Text	Function
E1	It rained heavily.	Reason for cancellation
E2	The match was cancelled.	Central event
E3	It will be played on Sunday instead.	Rescheduling detail

Step 2: Identify the first relation

E1 explains the cause of the cancellation in E2. Treat E2 as the nucleus and E1 as the satellite. The word “because” supports a **Cause** analysis, though the interpretation must also follow the meaning of the clauses.

Cause: Nucleus = E2; Satellite = E1

Step 3: Combine the larger discourse span

E3 adds information about the match after cancellation. A reasonable analysis attaches E3 as an **Elaboration** satellite to the span containing E1 and E2. Exact relation names and attachment choices can vary with the annotation scheme.

Elaboration(N = Cause(N = E2, S = E1), S = E3)

Step 4: Interpret the representation

The representation states that the cancellation is central, heavy rain explains why it happened, and Sunday provides a rescheduling detail. This is a hierarchical discourse analysis rather than a word-level syntactic parse.

Application: RST helps identify important content for summarization, assess text coherence, and explain how arguments and supporting information are organized.

5(b). Compare stemming and lemmatization using example words. (CO3-L3) (5M)

Ans:

Definitions

Stemming reduces a word to a stem by applying heuristic rules, often stripping suffixes. The result need not be a dictionary word.
Lemmatization maps an inflected word to its dictionary base form, called a lemma, using vocabulary and morphological analysis. It often requires part-of-speech information.

Comparison

Aspect	Stemming	Lemmatization
Method	Heuristic affix rules	Lexicon and morphological rules
Output	May be a nonword stem	Usually a dictionary lemma
POS / context	Usually not used	Often needed for correct analysis
Computation	Usually simpler and faster	Usually more linguistically informed
Typical use	Search indexing and retrieval	Linguistic analysis and meaning-sensitive NLP

Worked examples

The stems below are illustrative outputs of the **original Porter stemmer**. Lemmas assume appropriate POS tags; other algorithms and lexical resources may differ.

Word / POS	Porter stem	Lemma
studies / verb	studi	study
running / verb	run	run
connected / verb	connect	connect
mice / noun	mice	mouse
better / adjective	better	good

Step-by-step explanation

Stemming "studies": suffix rewriting yields "studi", useful as a matching key although it is not an English dictionary word.
Lemmatizing "mice": a lexical morphological rule recognizes the irregular plural and returns "mouse"; simple suffix removal cannot do this.
Lemmatizing "better": with adjective information and a suitable lexicon, the comparative form maps to "good".

Why POS matters

"Saw" as a past-tense verb may be lemmatized to **see**, while "saw" as a noun meaning a cutting tool remains **saw**. This illustrates why grammatical context can change the correct lemma.

Conclusion: Stemming is suitable when approximate word matching is sufficient. Lemmatization is preferable when a valid base form and the intended grammatical analysis matter.

---

6. Utilize Centering Theory to track discourse focus using forward-looking and backward-looking centers. (CO3-L3) (10M)

Ans:

Definition

Centering Theory models local discourse coherence by tracking the entities mentioned in successive utterances. It explains how discourse focus continues or shifts and provides constraints and preferences useful in pronoun resolution.

1. Forward-looking centers: Cf(U)

Cf(U) is an ordered list of the discourse entities realized in utterance U. These are candidates for the focus of a following utterance. A commonly used simplified ranking for English is **subject > object > other entities**. The precise ranking depends on language and analysis.

2. Preferred center: Cp(U)

Cp(U) is the highest-ranked entity in Cf(U). It represents the entity favored as the next discourse focus. Under the simplified ranking, it is usually the subject.

3. Backward-looking center: Cb(U)

Cb(U) is the highest-ranked member of **Cf(U-1)** that is realized in the current utterance U. It links the current utterance to the preceding one. It is not simply the current subject or the first repeated name.

$Cp(U_i)$ = highest-ranked entity in $Cf(U_i)$

$Cb(U_i)$ = highest-ranked entity in $Cf(U_{i-1})$ realized in U_i

Step 1: Read the discourse

U1: Ravi met Sita.
U2: He greeted her.
U3: Sita thanked him.
U4: She smiled.

For this worked analysis, assume “he/him” refers to Ravi and “she/her” refers to Sita. These links are contextually plausible and allow the focus calculations below.

Step 2: Compute Cf, Cp, and Cb

Utterance	Cf: ranked entities	Cp	Cb
U1	[Ravi, Sita]	Ravi	Undefined
U2	[Ravi, Sita]	Ravi	Ravi
U3	[Sita, Ravi]	Sita	Ravi
U4	[Sita]	Sita	Sita

U2: Both Ravi and Sita from U1 recur. Ravi ranks higher in Cf(U1), so Cb(U2) = Ravi.
U3: Both entities from U2 recur. Ravi ranks higher in Cf(U2), so Cb(U3) remains Ravi, even though Sita is now the subject and preferred center.
U4: Only Sita from Cf(U3) occurs, so Cb(U4) = Sita.

Step 3: Classify focus transitions

Transitions compare the current and preceding backward-looking centers, and whether the current backward-looking center equals the current preferred center.

Transition	Cb relation to previous Cb	Current Cb vs Cp
Continue	Same	Cb = Cp
Retain	Same	Cb differs from Cp
Smooth Shift	Different	Cb = Cp
Rough Shift	Different	Cb differs from Cp

The usual coherence preference is **Continue > Retain > Smooth Shift > Rough Shift**. This is a preference among possible analyses, not a rule that makes every shift incoherent.

Step 4: Apply the transition definitions

Point	Reasoning	Transition
U1 to U2	U1 has no previous Cb in this standalone discourse.	Initialization
U2 to U3	Cb remains Ravi, while Cp becomes Sita.	Retain
U3 to U4	Cb changes from Ravi to Sita and equals Cp(U4).	Smooth Shift

U2 establishes Ravi as the local center. U3 keeps Ravi as the backward link while promoting Sita to subject position. U4 completes the shift by making Sita both the backward-looking and preferred center. Some teaching conventions label an initial compatible transition as Continue; here it is kept separate because Cb(U1) is undefined.

Step 5: Use centering for pronoun interpretation

Consider **“Ravi met Sita. He smiled.”** Gender compatibility favors Ravi for “he”, and Ravi is also the highest-ranked entity in Cf(U1), giving a coherent continuation of focus. For **“Ravi met Mohan. He smiled.”**, both names may be compatible; the subject-ranking preference favors Ravi, but context can support Mohan instead.

A commonly stated centering constraint is: if any entity from Cf(U-1) is realized by a pronoun in U, then Cb(U) must also be pronominalized. This constrains preferred realizations; it does not eliminate all referential ambiguity.

Applications and limitations

Centering supports anaphora resolution, coherent text generation, dialogue management, and discourse analysis. It captures local entity continuity but does not by itself resolve world-knowledge questions, long-distance reference, or every ambiguous pronoun. Entity identification and grammatical ranking must be supplied by the analysis.

Result: Forward-looking centers identify potential next foci, the preferred center selects the most prominent current entity, and the backward-looking center records continuity with the preceding utterance.