REG.NO.:

SLOT: C1+TC1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

Programme Name & Branch : B.Tech Computer Science and Engineering
Course Code and Course Name : BCSE409L Natural Language Processing
Faculty Name(s) : Dr. SARAVANAKUMAR K, Dr. KANNADASAN R, Dr. RAJESHKANNAN R, Dr. SATHYARAJ R, Dr. JABANJALIN HILDA J, Dr. ARIVOLI A
Class Number(s) : VL2025260502034, 2035, 2032, 2033, 2039, 2270
Date of Examination : 29-01-2026
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

Answer All Questions

- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes
  CO1-Understand the fundamental concepts of Natural Language Processing.
  CO2-Develop useful systems for language processing and related tasks involving text processing and demonstrate text-based processing of natural language with respect to morphology.
  CO3-Check the syntactic and semantic correctness of natural language

Q. No	Question	Module	Marks	CO	BL
1.	A healthcare organization wants to use NLP to automatically generate short summaries from clinical reports to assist doctors in quick diagnosis. For example, from a medical note: *"The patient exhibits elevated blood glucose levels and persistent fatigue, indicating possible type-2 diabetes."* Explain in detail how the different stages of NLP will process this input to generate a concise medical summary. Solution: **Stages of NLP in Medical Text Summarization** **A. Text Acquisition / Input Processing** Clinical report is provided as raw text Input may come from: Electronic Health Records (EHR),Doctor's notes.Discharge summaries **B. Lexical Analysis (Tokenization)** Sentence is split into tokens (words and punctuation) The | patient | exhibits | elevated | blood | glucose | levels | and | persistent | fatigue | , | indicating | possible | type-2 | diabetes Purpose:Converts raw text into manageable units for processing Normalization & Cleaning-Converts text to a standardized form:- Lowercasing,Removing punctuation,Handling hyphenated terms Example: "type-2" → "type 2" "glucose levels" treated as a single medical phrase Stop-Word Removal-Removes non-informative words Removed words:the, and, indicating	1	10	CO1	BL2

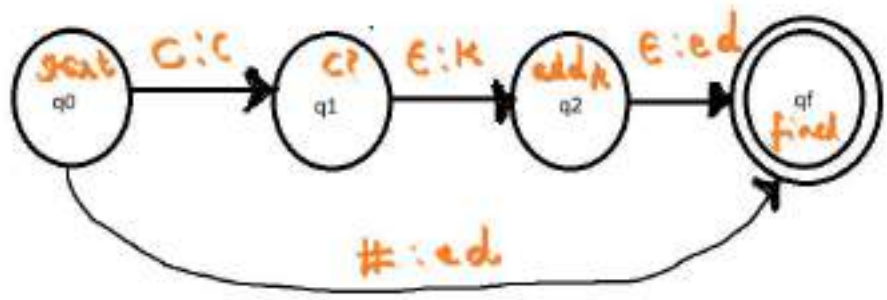
	Remaining keywords:patient, exhibits, elevated, blood glucose, persistent fatigue, possible, type-2 diabetes C. **Morphological Analysis (Stemming/Lemmatization)-**Reduces words to base forms Examples: exhibits → exhibit elevated → elevate indicating → indicate Benefit:Helps in identifying medical concepts accurately Part-of-Speech (POS) Tagging-Assigns grammatical roles to each word Examples: patient → Noun elevated → Adjective blood glucose → Noun Phrase diabetes → Noun Role in summarization: Helps identify: Sy-ptoms, Conditions, Medical entities Named Entity Recognition (NER)-Identifies medical entities Extracted entities: Symptom: persistent fatigue Lab indicator: elevated blood glucose Disease: type-2 diabetes Importance:-Crucial for clinical summarization and diagnosis assistance D. **Syntactic Analysis (Parsing)-**Analyzes sentence structure and relationships Understanding relationships:Elevated blood glucose + fatigue → indicate → possible type-2 diabetes Outcome:Establishes cause-effect and symptom-disease links E. **Semantic Analysis-**Interprets the meaning of the sentence, Uses medical ontologies Semantic inference: High blood glucose + fatigue = strong indicators of diabetes "possible" indicates uncertainty (important in clinical context) **F. Information Extraction** Selects only clinically relevant facts: Condition suspected Key symptoms Diagnostic indicators Extracted information: Elevated blood glucose, Persistent fatigue, Possible type-2 diabetes Text Summarization Uses extractive or abstractive summarization: Extractive: selects important phrases Abstractive: rewrites in concise medical language **Generated Summary**: *"Elevated blood glucose and fatigue suggest possible type-2 diabetes."* Post-Processing & Validation-Ensures: Medical accuracy, No loss of critical information, Concise format suitable for doctors G. **Final Output Summary-***"Elevated blood glucose and persistent fatigue indicate possible type-2 diabetes."*				

VIT
Vellore Institute of Technology

REG.NO.:

SLOT: C1+TC1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

2.	Identify the type(s) of ambiguity present in the following sentences and justify your answers with appropriate explanations. a) The teacher discussed the exam with the students in the hall. b) Anita saw the girl with the binoculars. c) Rahul informed Suresh that he had won the competition. d) Can you pass the salt? e) After Neha spoke to Riya, she said she would resign from the project. Solution: (a) Structural (Syntactic) (b) Structural (Attachment) (c) Referential (Anaphora) (d) Pragmatic (e) Referential (Anaphora)	1	10	CO1	BL2
3.	a) Design a Finite State Transducer (FST) to model the K-insertion rule for English verbs ending with –c when forming the past tense. i. Describe the states and input–output transitions of the FST. (2 marks) ii. Explain how the FST inserts 'k' only for verbs ending with –c and maps other verbs to their regular past-tense forms without insertion. (3 marks) *(Examples: mimic → mimicked, panic → panicked, traffic → trafficked, play → played)* *Solution:*  **State meanings** • q0 – reads and copies all characters • q1 – remembers that the last symbol read is c • q2 – inserts k • qF – final state **Transitions** • c:c → move to q1 • #:ed → normal past tense • ε:k → insert k(only after final c)	2	5	CO2	BL3

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

	• ε:ed → add past-tense suffix ✓ Verbs not ending in c go from q0 → qF with ed. ✓ Verbs ending in c go from q1 → q2 → qF, inserting k before ed. ✓ No other verb can reach the k-insertion path.				
	b) . Given the following mini corpus, Document 1: Dr. Sharma joined the AI lab in 2021. He published 3.5 papers per year on avg. Document 2: The system achieved an accuracy of 98.7%! Can it be improved further? Yes, it can. Explain the decision-tree-based sentence segmentation algorithm and show how it segments the above corpus. *(5 marks)*  Document 1: Dr. Sharma joined the AI lab in 2021. He published 3.5 papers per year on avg. a. **Dr.** → Abbreviation → **Not Boundary** b. **2021. He** → Not abbreviation + Not decimal + Next word capitalized → **Boundary** c. **3.5** → Decimal → **Not Boundary** d. **avg.** → End of text → **Boundary** **Segmented sentences** **S1:** Dr. Sharma joined the AI lab in 2021. **S2:** He published 3.5 papers per year on avg. **Document 2** The system achieved an accuracy of 98.7%! Can it be improved further? Yes, it can. a. 98.7 • Punctuation: . • Part of a decimal number • Decision-tree outcome: Not Boundary	2 5		CO2	BL3

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - I
WINTER SEMESTER 2025-2026

---

b. ! (after %)
- Punctuation: !
- Decision-tree rule: ! → Boundary

c. ? (after *further*)
- Punctuation: ?
- Decision-tree rule: ? → Boundary

d. . (after *can*)
- End of text → Boundary

Final segmented sentences

Sentence 1: The system achieved an accuracy of 98.7%!
Sentence 2: Can it be improved further?
Sentence 3: Yes, it can.

4. Given the following transition and emission probabilities, use the **Viterbi algorithm** to find the most probable tag sequence for the sentence:

"time flies like arrows"

Emission Probabilities	Transition Probabilities
P("time"|N)=0.30 P("flies"|V)=0.20 P("flies"|N)=0.05 P("like"|P)=0.40 P("arrows"|N)=0.60	P(V|N)=0.5 P(P|V)=0.4 P(N|P)=0.8 P(N|N)=0.2

The initial probabilities are:

$P(N) = P(V) = P(P) \approx 0.33$

Perform the following operations.

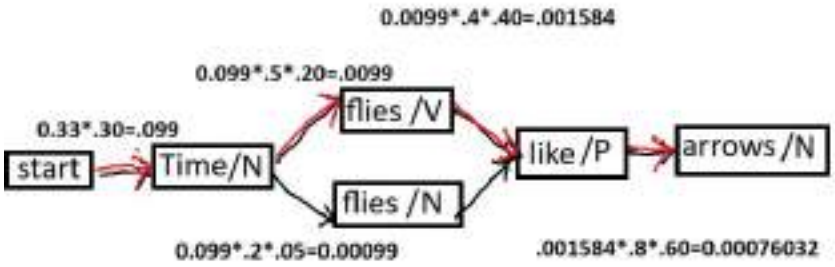
(i) Construct the Viterbi trellis
(ii) Show calculations at each step
(iii) Identify the best tag sequence

Solution:

Calculations:

Word 1: "time" N-> 0.33 × 0.30 = 0.099
: "time" V -> No emission given → 0
: "time" P->No emission given → 0
Therefore at *time*: N (0.099)
Word 2: "flies" :N From N → N
→0.099 × 0.2 × 0.05 = 0.00099
: V From N → V
→0.099 × 0.5 × 0.20 = 0.0099
Therefore at *flies*: V (0.0099)
(backpointer: N → V)
Word 3: "like": P From V → P
→0.0099 × 0.4 × 0.40 = 0.001584
Other tags have zero emission.

4.	2	10	CO2	BL3

	Therefore *like*: P (0.001584) (backpointer: V → P) Word 4: "arrows" N From P → N -→0.001584 × 0.8 × 0.60 = 0.00076032 Therefor at *arrows*: N (0.00076032) (backpointer: P → N) **Most probable tag sequence:** time flies like arrows N V P N 				
5.	A news aggregation company is building a rule-based NLP system to extract key phrases (Noun Phrases and Verb Phrases) from short news headlines. Because headlines are short, linear, and mostly follow fixed POS patterns, the development team decides to use a Finite State Transducer (FST) for Shallow parsing instead of a full syntactic parser. The system receives POS-tagged input and must output chunk labels. Consider the following headline: **"The skilled engineer designed a bridge"** The chunking rules followed by the system are: • *NP → (DET) (ADJ) N** • VP → V (NP)? Perform the following operations: i. Assign each token with the POS tag for the given sentence. ii. Design a Finite State Transducer (FST) for identifying Noun Phrase (NP) chunks. • Clearly mention the states, transitions, and accepting states. iii. Extend the FST to handle Verb Phrase (VP) chunking. iv. Apply the designed FST to the given POS-tagged sentence and generate the labels for each word and write the final shallow-parsed output by grouping words into chunks. v. Discuss the advantages and limitations of shallow parsing techniques and explain the applications of shallow parsing in NLP systems. *[0+3+2+2+3]* *Solution:* 1. **Shallow parsing (chunking)** using an **FST** Rules: **NP → (DET) (ADJ) N***	3	10	CO3	BL4

VP → V (NP)?

i. POS tagging of the sentence

The skilled engineer designed a bridge

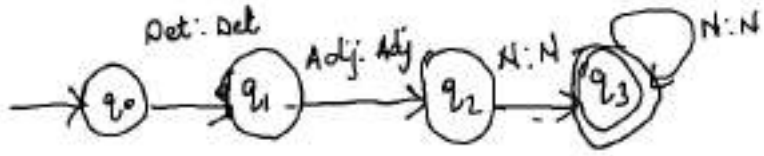
Word	POS tag	Reason
The	DET	Determiner
skilled	ADJ	Adjective
engineer	N	Noun
designed	V	Verb (past tense)
a	DET	Determiner
bridge	N	Noun

ii. FST for Noun Phrase (NP) identification

NP Rule

NP → (DET) (ADJ) N*

NP FST (diagram)



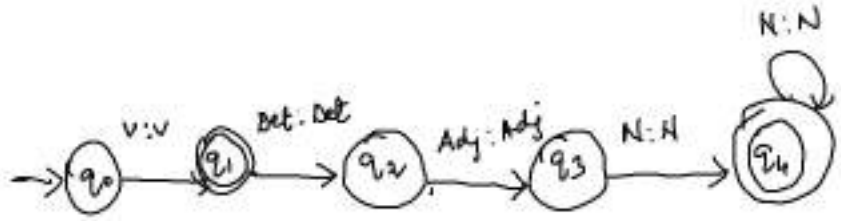
Accepting State

q3 (NP successfully recognized)

iii)FST to handle Verb Phrase (VP)

VP Rule

VP → V (NP)?



iv. Apply FST and generate shallow-parsed output

POS-tagged input

The/DET skilled/ADJ engineer/N designed/V a/DET bridge/N

Chunk identification

The skilled engineer

Matches: DET + ADJ + N

→ NP

designed

	Matches: V → start of VP a bridge Matches: DET + N → NP Combined with verb → VP **Final shallow-parsed output** [NP The skilled engineer] [VP designed [NP a bridge]] **v. Advantages, limitations, and applications of shallow parsing** Advantages Fast and efficient (linear time) No full parse tree required Well-suited for short, structured text (headlines, queries) Easy to implement using FSTs or regex-like rules Limitations Cannot capture deep syntactic relations Fails with complex or long sentences Limited handling of: Coordination Nested structures Long-distance dependencies Applications Information extraction Named Entity Recognition (NER) Question answering News headline analysis Preprocessing for full parsers Search engines and chatbots				

****