NLP WEEKEND QUESTIONS(3*10=30M)

First-Order Logic (FOL):

First-Order Logic (FOL), also called Predicate Logic, is a powerful formal system that extends Propositional Logic by introducing objects, relations, and quantifiers. It is essential in AI for knowledge representation and automated reasoning.

Components of FOL:

- **Constants**: Represent specific, fixed objects (e.g., John, Apple).
- **Variables**: General placeholders for objects (e.g., x, y, z).
- **Predicates**: Describe properties of objects and relationships between them (e.g., Human(x), Likes (John, Pizza)).
- **Functions**: Map objects to other objects (e.g., Father (John)).
- **Quantifiers**:
    - **Universal Quantifier (∀)**: Applies a statement to all objects

      (e.g., ∀x (Human(x) → Mortal(x))).
    - **Existential Quantifier (∃)**: States that a statement applies to at least one object

      (e.g., ∃x (Likes (x, Pizza))).

FOL is more expressive than Propositional Logic and can represent complex knowledge involving objects and their relationships.

Inference in First-Order Logic: Inference is the process of deriving new information from known facts and rules. In AI, inference automates reasoning and helps systems draw logical conclusions.

Why Inference is Needed:

- Automates logical reasoning in AI systems.
- Draws conclusions from general rules and specific facts.
- Handles complex object relationships efficiently.

Types of Inference in FOL:

- **Forward Chaining**: Starts from known facts and applies inference rules to derive new facts step by step.
- **Backward Chaining**: Starts from a goal and works backward to find supporting facts.
- **Resolution**: A powerful method based on proof by contradiction.

Propositional Vs. First Order Inference

Feature	Propositional Logic	First-Order Logic
Expressiveness	Simple facts (P → Q)	Describes objects & relations (∀x (Human(x) → Mortal(x)))
Scalability	Limited	Handles real-world complexity
Variables/Objects	None	Uses variables & quantifiers
Example	Rain → WetGround	∀x (Human(x) → Mortal(x))

Types of Inferences

1. Unification:

Unification is the process of making two logical expressions identical by substituting variables. It is a key concept in automated reasoning.

Example: Given:

- Knows (x, y)
- Knows (John, z)

Unification:

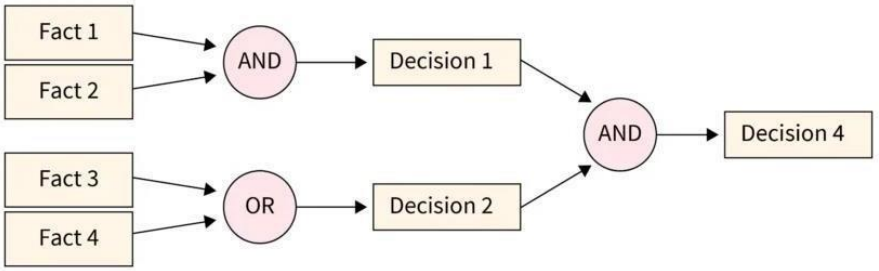
- x = John
- y = z

2. Lifted Inference:

- **Definition:** Uses **variables and quantifiers directly** without **instantiating specific constants**.
- **Benefit:** More **efficient reasoning** over general rules.
- **How Lifted Inference Works:** Instead of substituting each student one by one:
  - **Alice → LovesArtificialIntelligence (Alice)**
  - **Bob → LovesArtificialIntelligence (Bob)**
  - **Charlie → LovesArtificialIntelligence (Charlie)**
- Lifted inference **keeps the variable** and applies the rule directly:
  - From ∀**x (Student(x) → LovesArtificialIntelligence (x))**,
- We infer that **for all x where Student(x) is true, LovesArtificialIntelligence (x) is also true**.

3. Forward Chaining (Data-Driven):

Forward Chaining starts with facts and applies inference rules to derive new facts.

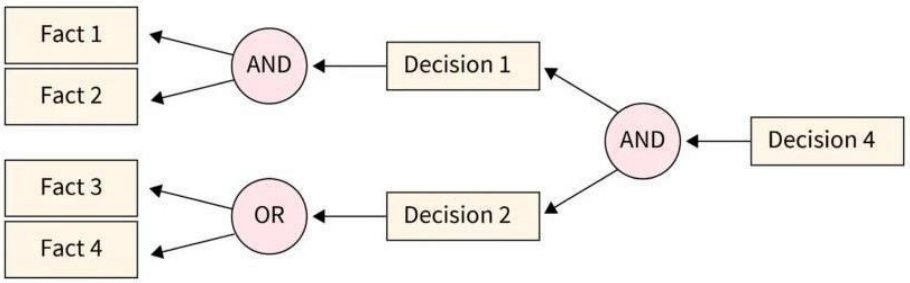


Example:

- Fact: Human(Socrates)
- Rule: ∀x (Human(x) → Mortal(x))
- Inference: Mortal(Socrates)

4. Backward Chaining (Goal-Driven):

Backward Chaining starts with a goal and works backward to find supporting facts.



Example:

- Goal: Mortal(Socrates)
- Rule: ∀x (Human(x) → Mortal(x))
- Fact: Human(Socrates)

5. Resolution (Proof by Contradiction):

Resolution is a powerful inference method based on proof by contradiction.

Steps:

1. Convert statements to clausal form.
2. Apply unification.
3. Use resolution to derive a contradiction.
4. Real-Time Advanced Example: Employee Promotion System Facts:

- HasSkill(John, Python)
- HasExperience(John, 5Years)
- HasGoodReviews(John)

Rule: ∀x (HasSkill(x, Python) ∧ HasExperience(x, ≥3Years) ∧ HasGoodReviews(x) → EligibleForPromotion(x))

Inference: EligibleForPromotion(John)

2. Word senses and relations between word senses

Word Senses in NLP

- In NLP, a word sense is the meaning of a word in a specific context.
- Since many words are ambiguous (polysemy/homonymy), an NLP system must decide which sense is intended.
- This process is called Word Sense Disambiguation (WSD).

Example:

- Sentence 1: *"I went to the bank to deposit money."* → Bank = financial institution.
- Sentence 2: *"He sat on the river bank."* → Bank = river side.

Relations Between Word Senses in NLP

NLP uses **semantic relations** (stored in resources like **WordNet**) to represent and process meaning.

1. Synonymy → Words with similar meaning.

Example: *big ≈ large*

Use: Query expansion in **search engines**.

2. Antonymy → Words with opposite meaning.

Example: *hot* ↔ *cold*

Use: **Sentiment analysis**, opinion mining.

3. Hyponymy/Hypernymy (IS-A hierarchy)

Hyponym: *Dog* is a hyponym of *Animal*.

Hypernym: *Animal* is a hypernym of *Dog*.

Use: **Ontology building**, taxonomy for knowledge graphs.

4. Meronymy/Holonymy (Part-whole relation)

Meronym: *Wheel* is part of a *Car*. Holonym: *Car* has *Wheel*.

Use: **Information extraction**, semantic parsing.

5. Polysemy → Word with multiple related senses.

Example: *Head* (body part / leader).

Use: **Word Sense Disambiguation (WSD)**.

6. Homonymy → Same spelling/sound, unrelated meanings.

Example: *Bat* (animal / cricket bat).

Use: **Speech recognition** & **context resolution**.

7. Troponymy (Verbs specialization)

Example: *Stroll* is a troponym of *Walk*.

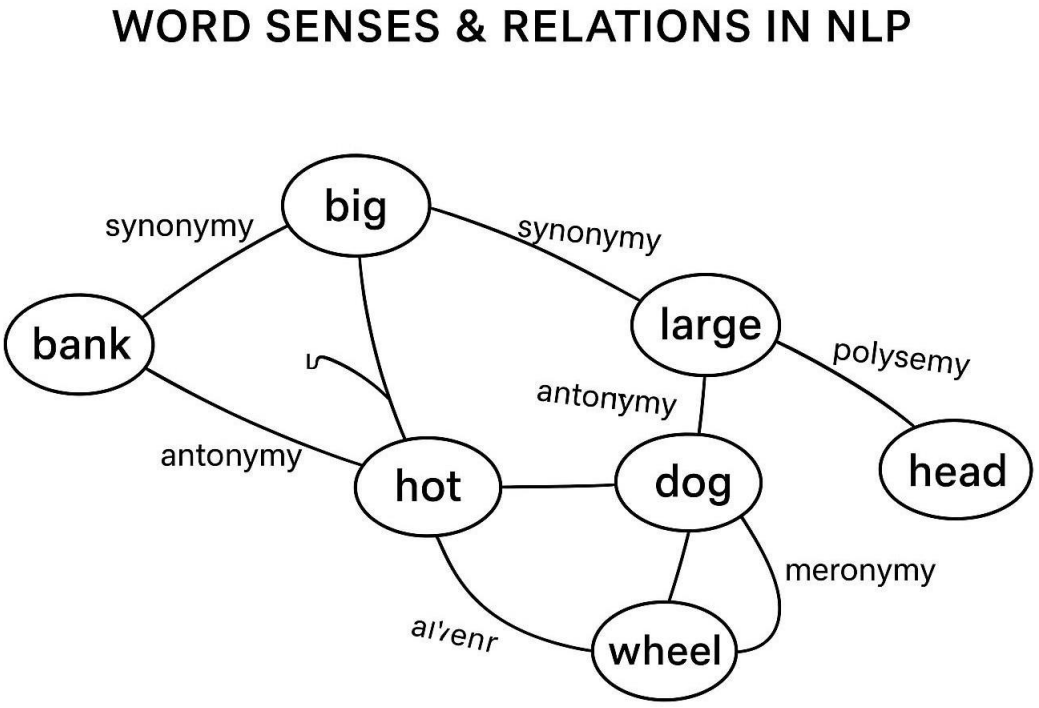
Use: **Action recognition in NLP**.

Applications:

Machine Translation → Correct word sense ensures

accurate translation.

- **Chatbots & QA Systems** → Understanding user queries correctly.
- **Information Retrieval/Search Engines** → Expands queries with synonyms/hypernyms.
- **Text Summarization & Generation** → Captures meaning instead of just words.
- **Knowledge Graphs & Semantic Web** → Structured representation of relations.



3. Word Senses Disambiguity

Definition

Word Sense Disambiguation is the process of **identifying which sense (meaning) of a word is used in a given context**. Since many words in natural languages are **polysemous** (having multiple meanings), disambiguating them is crucial for accurate understanding and processing.

— Example:

- *“He sat on the river **bank**.”* → Bank = riverside
- *“She went to the **bank** to deposit money.”* → Bank = financial institution

---

🔹 Importance in NLP

- **Machine Translation** → Correct word sense ensures accurate translation into other languages.
- **Information Retrieval/Search Engines** → Improves search accuracy by matching intended meanings.
- **Chatbots and Virtual Assistants** → Enhances dialogue understanding and context awareness.
- **Sentiment Analysis** → Helps detect the correct tone or meaning of words.
- **Text Summarization & Question Answering** → Ensures relevant information is extracted.

🔹 Approaches to WSD

1. **Knowledge-based Approaches**
    - Use dictionaries, thesauri, and lexical databases like **WordNet**.
    - Example: Lesk Algorithm → Chooses the sense with the most word overlap between dictionary definitions and sentence context.
2. **Supervised Machine Learning Approaches**
    - Train classifiers (e.g., Naïve Bayes, Decision Trees, Neural Networks) on labeled data.

    - Requires large annotated corpora.

3. **Unsupervised Approaches**
    - Use clustering and statistical methods without labeled data.
    - Groups word occurrences into clusters representing senses.
4. **Hybrid Approaches**
    - Combine knowledge-based and machine learning methods for better accuracy.

🔹 Challenges in WSD

- **Ambiguity of natural language** (words may have subtle, context-dependent meanings).
- **Data sparsity** → Lack of large annotated datasets for training.
- **Domain dependence** → Word senses vary across domains (e.g., *cell* in biology vs. telecom).
- **Computational complexity** for large-scale systems.

🔹 Conclusion

Word Sense Disambiguation is a **core problem in NLP** that directly impacts the performance of higher-level tasks like translation, search, and dialogue systems. Although several approaches exist, it remains a challenging area due to the richness and variability of human language.