UNIT-I

1. What is morphological analysis in NLP?

Morphological analysis studies word structures by breaking them into morphemes, the smallest meaning-bearing units. It helps in tasks like stemming, lemmatization, and part-of-speech tagging. This analysis is crucial for handling inflectional and derivational word forms, enabling better text normalization, improving search accuracy, and enhancing NLP applications such as machine translation and information retrieval.

2. Explain the importance of word structure in NLP.

Word structure plays a vital role in NLP by enabling accurate text processing. Understanding morphemes improves tokenization, speech recognition, and machine translation. It helps in reducing vocabulary size, enhancing word embeddings, and improving syntactic parsing. Proper word structure analysis ensures that NLP models grasp meaning variations, contributing to better context understanding and more precise linguistic modeling.

3. What are morphemes, and how are they classified?

Morphemes are the smallest units of meaning in a language. They are classified into **free morphemes**, which can stand alone as words (e.g., "book"), and **bound morphemes**, which need attachment to other morphemes (e.g., "-ed" in "walked"). Bound morphemes are further divided into **inflectional** (e.g., plural "-s") and **derivational** (e.g., "-ness" in "kindness") morphemes.

4. Differentiate between inflectional and derivational morphology.

Inflectional morphology deals with grammatical variations without changing the word's core meaning or category (e.g., "walk" → "walked"). Derivational morphology, however, forms new words by adding prefixes or suffixes that change meaning or part of speech (e.g., "happy" → "happiness"). Inflectional changes maintain word identity, while derivational changes create new words with different lexical functions.

5. What are the challenges in morphological analysis?

Challenges in morphological analysis include handling irregular word formations, processing languages with rich inflection (e.g., Turkish, Finnish), resolving ambiguity in morpheme segmentation, and managing compounding. Some words have multiple possible interpretations, requiring context-sensitive models. Computational complexity also increases with highly inflected or agglutinative languages, demanding sophisticated algorithms like deep learning and rule-based hybrid approaches.

6. What is stemming, and how does it differ from lemmatization?

Stemming reduces words to their root forms by removing suffixes, often leading to inaccurate root words (e.g., "running" → "run" or "runn"). Lemmatization, in contrast, maps words to their base dictionary forms using linguistic analysis (e.g., "running" → "run"). Stemming is faster but less accurate, while lemmatization provides correct word forms, improving NLP applications such as search engines and sentiment analysis.

7. Explain the role of morphological models in text processing.

Morphological models help process text by analyzing word structure, improving tokenization, and reducing vocabulary size. These models support tasks like spelling correction, information retrieval, and sentiment analysis. By identifying morphemes, they improve search relevance, text summarization, and named entity recognition, enhancing the accuracy of NLP applications such as chatbots and translation systems.

8. How do morphological models handle compound words?

Morphological models handle compound words by segmenting them into their meaningful components. In English, compounds like "sunlight" are split into "sun" and "light." Some languages, like German, have complex compound structures requiring advanced decomposition techniques. Proper segmentation improves search engines, machine translation, and text comprehension, reducing ambiguity and enhancing linguistic modeling.

9. What is a finite-state transducer in morphology?

A finite-state transducer (FST) is an automaton used in computational morphology to model word formation. It processes input words and maps them to possible morphemes, enabling efficient text processing. FSTs are widely used in stemming, lemmatization, and phonological analysis, allowing NLP systems to handle inflections and derivations with rule-based approaches.

10. Explain the complexity of morphological approaches in NLP.

Morphological approaches are complex due to irregular word formations, ambiguous morphemes, and varying language structures. Some languages have simple morphology (e.g., English), while others are highly inflected (e.g., Finnish). Handling such diversity requires advanced models, including statistical learning, rule-based systems, and deep neural networks, to ensure efficient and accurate word analysis.

11. What are rule-based and statistical morphological analyzers?

Rule-based analyzers use predefined linguistic rules for morpheme segmentation, such as suffix-stripping algorithms. They are interpretable but rigid. Statistical analyzers, like Hidden Markov Models (HMMs) and deep learning-based models, use probabilities to infer morphological structures from large corpora. They are more flexible and adaptive but require extensive training data for high accuracy.

12. How do unsupervised methods assist in morphological analysis?

Unsupervised methods, such as clustering and deep learning, automatically identify morpheme patterns from unannotated data. These methods analyze word co-occurrences and phonetic similarities to infer word structures without predefined rules. They are beneficial for low-resource languages where annotated datasets are scarce, improving NLP models' adaptability to new languages.

13. What is the role of morphology in speech recognition?

Morphology improves speech recognition by enhancing word segmentation and pronunciation modeling. It helps speech-to-text systems recognize different word inflections, reducing errors in transcription. By understanding morphemes, NLP models can accurately process spoken words, improving voice assistants, transcription services, and automatic subtitling in various languages.

14. How does morphology impact machine translation?

Morphology plays a crucial role in machine translation by ensuring accurate word mapping between languages. Morphological models help systems understand inflected forms, enabling better translation accuracy. Languages with rich morphology require sophisticated techniques like subword tokenization, improving the quality of translations in neural machine translation models like Google Translate.

15. What is the significance of morphological disambiguation?

Morphological disambiguation resolves ambiguity in word segmentation and meaning. Words like "leaves" (plural noun or verb) require context to determine their function. Statistical and deep-learning models analyze surrounding words to assign correct morphemes, improving NLP tasks like part-of-speech tagging, syntactic parsing, and text understanding.

UNIT – 2

1. **What is syntax analysis in NLP?**
   Syntax analysis, or parsing, examines sentence structure based on grammar rules. It helps identify relationships between words, ensuring proper sentence formation. Parsing techniques like dependency parsing and constituency parsing are used in machine translation, speech recognition, and text summarization, enabling NLP systems to derive meaningful syntactic structures from textual data.
2. **What is a parse tree, and why is it important?**
   A parse tree represents the syntactic structure of a sentence, breaking it into hierarchical components. It is essential in NLP for understanding sentence grammar, aiding in tasks like syntax-based machine translation, sentiment analysis, and question-answering systems. Parse trees help machines interpret human language more effectively by enforcing grammatical correctness.
3. **What are treebanks, and how are they used in NLP?**
   Treebanks are linguistic databases containing manually annotated syntactic structures of sentences. They serve as training datasets for supervised learning models in NLP tasks like parsing and language modeling. Popular treebanks, such as the Penn Treebank, provide standardized syntactic annotations that enhance the accuracy of machine learning-based syntactic analysis.
4. **Explain the difference between dependency parsing and constituency parsing.**
   Dependency parsing establishes relationships between words based on dependencies, forming a directed graph. Constituency parsing, however, breaks sentences into hierarchical phrase structures using phrase-structure rules. While dependency parsing focuses on word relationships, constituency parsing provides a broader syntactic breakdown, making both essential for different NLP applications.
5. **What is a context-free grammar (CFG)?**
   A context-free grammar (CFG) consists of production rules that define sentence structures. It includes a set of non-terminals, terminals, and a start symbol. CFGs are widely used in NLP for parsing, enabling applications like speech recognition and compiler design to analyze and process language systematically.
6. **What are parsing algorithms, and why are they important?**
   Parsing algorithms analyze sentence structures based on grammar rules. Examples include CYK (Cocke-Younger-Kasami) and Earley’s algorithm. They are crucial for syntactic analysis in NLP applications like chatbots, information extraction, and machine translation, ensuring that sentences follow correct grammatical structures for meaningful interpretation.
7. **What are top-down and bottom-up parsing approaches?**
   Top-down parsing starts with the highest-level grammar rules and recursively breaks them into smaller components, like in recursive descent parsing. Bottom-up parsing, on the other hand, begins with input tokens and builds up the parse tree, as seen in shift-reduce parsing. Both methods are used in syntax analysis.
8. **What is an ambiguity in parsing, and how is it handled?**
   Ambiguity occurs when a sentence has multiple valid parse trees, making interpretation

difficult. Ambiguities are handled using probabilistic parsing, disambiguation techniques, or additional contextual information. Lexical and syntactic ambiguities challenge NLP applications, requiring sophisticated models for resolution.

9. **What is lexical ambiguity in NLP?**
   Lexical ambiguity occurs when a word has multiple meanings, such as "bank" referring to a financial institution or a riverbank. NLP models resolve lexical ambiguity using word sense disambiguation techniques like knowledge-based methods, supervised learning, or contextual embeddings from models like BERT.
10. **How does syntactic ambiguity affect sentence parsing?**
    Syntactic ambiguity arises when a sentence has multiple syntactic interpretations, affecting NLP tasks like translation and speech recognition. For example, "She saw the man with a telescope" can mean different things. Disambiguation is achieved using statistical models, parse scoring, or contextual understanding.
11. **What are probabilistic context-free grammars (PCFGs)?**
    Probabilistic Context-Free Grammars (PCFGs) assign probabilities to grammar rules, improving parsing accuracy by choosing the most probable parse tree. PCFGs enhance NLP applications like speech recognition and translation by refining syntax analysis based on training data probabilities.
12. **What is the role of feature-based grammars in parsing?**
    Feature-based grammars, such as Lexical Functional Grammar (LFG) and Head-Driven Phrase Structure Grammar (HPSG), incorporate additional linguistic features like tense, gender, and agreement. They improve syntactic parsing by integrating richer linguistic constraints, enhancing NLP models for better language understanding.
13. **How does syntax analysis improve machine translation?**
    Syntax analysis helps machine translation by ensuring grammatical accuracy, improving word alignment, and handling sentence structure variations across languages. Syntax-based approaches, such as hierarchical phrase-based translation and syntactic transfer models, enhance translation quality by maintaining linguistic integrity.
14. **What are multilingual issues in parsing?**
    Multilingual parsing challenges include language-specific grammar rules, word order variations, and lack of annotated treebanks for low-resource languages. Cross-lingual parsing techniques, such as transfer learning and multilingual embeddings, help address these issues, making NLP models more adaptable to diverse languages.
15. **What are transition-based dependency parsers?**
    Transition-based dependency parsers construct parse trees by performing a sequence of actions on an input sentence. Algorithms like the Arc-standard and Arc-eager parsing methods efficiently determine syntactic dependencies, making them useful for real-time applications like speech recognition and chatbots.

UNIT – 3

1. **What is semantic parsing in NLP?**
   Semantic parsing converts natural language into a structured representation that machines can understand. It helps in extracting meaning from text by mapping sentences to logical forms or database queries. Applications include virtual assistants, question-answering systems, and knowledge graphs. Semantic parsing is crucial for improving the accuracy of natural language understanding in AI systems.
2. **What are the key challenges in semantic parsing?**
   Challenges in semantic parsing include handling ambiguous sentences, understanding context, resolving word sense disambiguation, and managing complex sentence structures. Additionally, domain adaptation and training high-quality models require large annotated datasets. Developing models that generalize across different languages and contexts remains a major difficulty in semantic parsing research.
3. **What is the difference between syntactic and semantic parsing?**
   Syntactic parsing focuses on grammatical structure, identifying sentence components like nouns and verbs. Semantic parsing, however, extracts meaning by mapping text to structured representations, such as logical forms. While syntactic parsing ensures grammatical correctness, semantic parsing enables deeper understanding for applications like chatbots, question answering, and machine translation.
4. **Explain the role of semantic interpretation in NLP.**
   Semantic interpretation helps convert text into meaningful representations by analyzing sentence structure and word relationships. It is essential for accurate information extraction, sentiment analysis, and contextual understanding in applications like virtual assistants and text summarization. By understanding meanings beyond syntax, semantic interpretation improves machine comprehension of human language.
5. **What are system paradigms in semantic parsing?**
   System paradigms in semantic parsing include rule-based approaches, statistical models, and deep learning-based methods. Rule-based systems rely on predefined grammar, statistical methods use probabilistic models, and deep learning leverages neural networks to improve accuracy. Each approach has strengths and limitations, depending on data availability and complexity.
6. **What is the role of semantic parsing in question-answering systems?**
   Semantic parsing enhances question-answering systems by converting questions into structured queries for database retrieval. It helps understand user intent, resolve ambiguities, and provide accurate responses. By mapping natural language to logical forms, semantic parsing ensures precise and relevant answers in AI-driven applications like chatbots and search engines.
7. **How does word sense disambiguation (WSD) aid semantic parsing?**
   WSD helps determine the correct meaning of words with multiple interpretations by analyzing context. It improves semantic parsing accuracy by ensuring proper mapping of words to their intended meanings. Techniques include knowledge-based, supervised, and unsupervised methods, aiding applications like machine translation, search engines, and text classification.
8. **What is the importance of compositional semantics in NLP?**
   Compositional semantics builds sentence meaning from individual word meanings and

their syntactic relationships. It follows the principle that sentence meaning is a combination of its parts. This approach improves understanding in tasks like translation, sentiment analysis, and conversational AI, ensuring coherent and contextually accurate interpretations.

9. **Explain the role of deep learning in semantic parsing.**
   Deep learning enhances semantic parsing by leveraging neural networks for automatic feature extraction and pattern recognition. Models like transformers and recurrent neural networks (RNNs) improve accuracy in natural language understanding, reducing dependency on handcrafted rules. Deep learning enables robust parsing in applications like virtual assistants and automated reasoning systems.
10. **What are the applications of semantic parsing?**
    Semantic parsing is used in chatbots, virtual assistants, search engines, and machine translation. It improves question-answering, knowledge retrieval, and automated reasoning by enabling machines to understand and process human language effectively. Businesses leverage semantic parsing for customer support automation, data extraction, and personalized recommendations.
11. **What are the different techniques used in semantic parsing?**
    Techniques in semantic parsing include rule-based approaches, statistical models, and deep learning-based methods. Rule-based methods use predefined linguistic rules, statistical models rely on probabilistic parsing, and deep learning models utilize neural networks like transformers for better generalization. Each technique has specific use cases based on complexity and dataset size.
12. **How do knowledge graphs improve semantic parsing?**
    Knowledge graphs enhance semantic parsing by providing structured data for better contextual understanding. They help in entity recognition, relation extraction, and disambiguation, improving the accuracy of AI models. By connecting concepts, knowledge graphs support applications like search engines, recommendation systems, and automated reasoning.
13. **What is the significance of embedding techniques in semantic parsing?**
    Embedding techniques like Word2Vec, GloVe, and BERT represent words as dense vectors, capturing semantic relationships. These embeddings improve semantic parsing by providing context-aware representations, enhancing NLP tasks like sentiment analysis, translation, and question-answering. Advanced models like transformers further refine parsing accuracy through deep contextual embeddings.
14. **How does multilingual semantic parsing work?**
    Multilingual semantic parsing processes text across multiple languages, requiring robust cross-lingual embeddings and transfer learning techniques. Models like mBERT and XLM handle diverse linguistic structures, enabling better cross-language understanding. This capability is essential for global applications like multilingual chatbots and automated translation systems.
15. **What is the future of semantic parsing in NLP?**
    The future of semantic parsing involves advancements in deep learning, zero-shot learning, and self-supervised training. Improved models will enhance contextual understanding, enabling more accurate conversational AI, real-time translation, and intelligent search systems. Integration with cognitive computing and knowledge graphs will further refine NLP applications.

UNIT – 4

1. **What is Predicate-Argument Structure in NLP?**
   Predicate-Argument Structure represents the relationship between a verb (predicate) and its associated entities (arguments). It helps identify semantic roles, such as subject, object, and indirect object, within a sentence. This structure is essential for tasks like machine translation, information extraction, and question answering, as it captures deeper linguistic meaning beyond syntactic parsing.
2. **How does Predicate-Argument Structure enhance natural language understanding?**
   By identifying how verbs relate to nouns, Predicate-Argument Structure helps models understand sentence semantics more accurately. It supports applications like semantic role labeling (SRL), event extraction, and machine translation by structuring meaning in a way that machines can process, allowing better context recognition and reasoning in NLP systems.
3. **What are the key challenges in analyzing Predicate-Argument Structures?**
   Challenges include handling ambiguities in argument roles, dealing with implicit arguments, and resolving syntactic variations across languages. Complex sentence structures, idiomatic expressions, and missing arguments make automatic parsing difficult. NLP models require large annotated datasets and robust algorithms to improve performance in real-world applications.
4. **What are Meaning Representation Systems in NLP?**
   Meaning Representation Systems (MRS) encode the semantic content of sentences in a structured form. They transform natural language into logical representations, enabling reasoning, question answering, and inference. Examples include First-Order Logic (FOL), Abstract Meaning Representation (AMR), and Frame Semantics, which help in building AI-driven applications with deeper language understanding.
5. **Why is Meaning Representation important in NLP?**
   Meaning Representation allows computers to process and manipulate language meaningfully. It enables tasks like knowledge representation, automatic reasoning, and dialogue understanding. By converting text into formal logic or structured formats, NLP systems can infer relationships, detect contradictions, and generate accurate responses in AI applications.
6. **What are the different types of Meaning Representation Systems?**
   Common types include First-Order Logic (FOL), which uses predicates and variables for reasoning; Abstract Meaning Representation (AMR), which represents sentence meaning as graphs; and Frame Semantics, which structures meaning based on conceptual roles. Each approach has its advantages depending on the application and complexity of language understanding required.

7. **What is First-Order Logic (FOL) in NLP?**
   First-Order Logic (FOL) represents relationships between entities using logical predicates, quantifiers, and variables. It enables reasoning and inference in NLP applications such as knowledge representation and automated reasoning. FOL is

commonly used in AI-driven chatbots, expert systems, and question-answering models for structured decision-making.

8. **Explain Abstract Meaning Representation (AMR) in NLP.**
   AMR is a semantic representation that captures the meaning of a sentence as a directed graph. Nodes represent concepts, and edges define relationships. AMR abstracts away from syntax, focusing on underlying semantics, making it useful for machine translation, text summarization, and information retrieval applications.
9. **What is Frame Semantics, and how does it contribute to NLP?**
   Frame Semantics is a theory where words are understood within conceptual frames. A frame consists of roles and relationships that define a scenario. NLP applications use FrameNet, a database of frames, to enhance text understanding, question answering, and information extraction by linking words to real-world contexts.
10. **How does FrameNet assist in Meaning Representation?**
    FrameNet provides a structured linguistic database linking words to conceptual frames. It helps in tasks like semantic role labeling, information retrieval, and dialogue systems by associating words with their meanings in context, improving NLP applications' interpretability and accuracy.
11. **What role does Semantic Parsing play in Meaning Representation?**
    Semantic Parsing converts natural language into machine-readable formats, such as logical forms or structured graphs. It enables systems to understand, process, and generate meaningful responses in AI applications like chatbots, automated reasoning, and knowledge-based systems.
12. **What are the challenges in designing Meaning Representation Systems?**
    Challenges include handling ambiguity, representing implicit meanings, scaling across different languages, and integrating contextual understanding. Developing large annotated datasets, efficient algorithms, and computational resources is crucial for improving meaning representation in NLP applications.
13. **What are some real-world applications of Meaning Representation?**
    Applications include machine translation, automated reasoning, semantic search, text summarization, and AI-driven chatbots. By structuring language into logical representations, NLP models can understand context better and generate more relevant responses in various domains.
14. **How do Software Systems use Meaning Representation in NLP?**
    NLP software systems use Meaning Representation for tasks like knowledge extraction, automated dialogue systems, and intelligent search engines. Examples include IBM Watson, Google’s BERT, and OpenAI’s GPT, which utilize structured meaning to improve response accuracy and user interactions.
15. **What are the future trends in Meaning Representation for NLP?**
    Future trends include integrating deep learning with symbolic reasoning, improving multilingual meaning representation, and developing more explainable AI models. Advances in knowledge graphs, hybrid NLP approaches, and contextual embeddings will enhance semantic understanding in AI-driven applications.

UNIT – 5

1. What is discourse processing in NLP?

Discourse processing studies how sentences connect to form meaningful text. It includes analyzing coherence, reference resolution, and discourse structure. NLP applications, such as summarization, question answering, and dialogue systems, benefit from discourse analysis by understanding text relationships, pronoun resolution, and implicit meanings, improving machine comprehension and interaction.

2. What is discourse cohesion?

Discourse cohesion ensures logical text connectivity using linguistic elements like pronouns, conjunctions, and lexical repetition. It helps maintain consistency, making texts readable and meaningful. Cohesion techniques include reference substitution, ellipsis, and linking words, aiding NLP tasks such as summarization, machine translation, and sentiment analysis by improving textual structure.

3. Explain reference resolution in NLP.

Reference resolution involves linking pronouns, definite nouns, or implicit expressions to their referents within a text. It is crucial for anaphora and coreference resolution. Algorithms like rule-based systems, machine learning, and deep learning help improve accuracy. Reference resolution enhances chatbots, text summarization, and sentiment analysis by maintaining coherence.

4. What is discourse cohesion, and why is it important?

Discourse cohesion connects sentences logically using cohesive devices like conjunctions, synonyms, and pronouns. It enhances text readability, ensuring smooth transitions between ideas. Cohesion is crucial for NLP tasks like machine translation, summarization, and document retrieval, as it helps maintain structured, meaningful communication and reduces ambiguity in text interpretation.

5. What is discourse coherence, and how does it differ from cohesion?

Discourse coherence refers to logical consistency, ensuring sentences relate meaningfully. Unlike cohesion, which focuses on surface linguistic links, coherence relies on deeper semantic and pragmatic relationships. NLP applications like dialogue systems and summarization require coherence analysis to ensure texts are contextually appropriate and logically connected for user understanding.

6. Explain the role of discourse analysis in NLP applications.

Discourse analysis improves NLP applications by enhancing text understanding, summarization, sentiment analysis, and machine translation. It helps resolve ambiguities, identify discourse structure, and improve chatbot interactions. Techniques like coherence modeling and reference resolution ensure better context retention, leading to more accurate and natural text generation and processing.

7. What are N-Gram models in language modeling?

N-Gram models predict the probability of word sequences based on prior words. They are useful in speech recognition, text prediction, and machine translation. While simple and effective, they face limitations like sparsity and lack of long-term dependencies, which deep learning models like RNNs and transformers help overcome.

8. How does language model evaluation work?

Language model evaluation measures a model's ability to generate meaningful and grammatically correct text. Metrics like perplexity, BLEU score, and ROUGE assess performance. Lower perplexity indicates better fluency, while BLEU measures translation accuracy. Evaluation ensures improved machine translation, chatbots, and text summarization by refining NLP models.

9. What is parameter estimation in language models?

Parameter estimation determines probability distributions in NLP models. It involves statistical methods like Maximum Likelihood Estimation (MLE) and smoothing techniques to handle sparse data. Accurate estimation improves text prediction, speech recognition, and machine translation by refining word sequence probabilities and contextual understanding in language modeling.

10. Explain language model adaptation.

Language model adaptation modifies pre-trained models for domain-specific applications by fine-tuning on specialized datasets. Techniques like transfer learning, domain adaptation, and data augmentation improve accuracy in areas like medical NLP, legal text processing, and customer service chatbots, ensuring better contextual understanding and relevancy in specialized fields.

11. What are the types of language models?

Language models include N-Gram models, Hidden Markov Models (HMMs), Neural Network-based models (RNNs, LSTMs, Transformers), and probabilistic models. N-Grams use word frequency, HMMs rely on probabilistic sequences, while neural models leverage deep learning for contextual understanding. These models enhance speech recognition, translation, and text generation tasks.

12. What are language-specific modeling problems in NLP?

Language-specific modeling issues include handling inflections, complex syntax, polysemy, and low-resource languages. Some languages lack sufficient training data, requiring techniques like transfer learning. Word segmentation in Chinese, morphological complexity in Arabic, and agglutination in Turkish create unique challenges for NLP model training and optimization.

13. How does multilingual language modeling work?

Multilingual models process multiple languages using shared embeddings and pre-trained architectures like mBERT or XLM-R. These models enhance translation, cross-lingual retrieval, and sentiment analysis by learning language similarities. Challenges include maintaining consistency, handling syntax variations, and addressing low-resource language limitations in NLP applications.

14. What is cross-lingual language modeling?

Cross-lingual language modeling enables NLP systems to transfer knowledge between languages without direct training on all of them. Techniques like zero-shot and few-shot learning allow models to process underrepresented languages. This approach benefits multilingual chatbots, automatic translation, and knowledge transfer between different linguistic datasets.

15. What challenges arise in multilingual and cross-lingual NLP?

Challenges include handling different scripts, grammatical structures, idiomatic expressions, and data scarcity. Transliteration, translation inaccuracies, and limited training datasets affect model performance. Advancements in transformer-based architectures like mBERT and fine-tuning strategies improve multilingual NLP accuracy while addressing linguistic diversity and complexity issues.