UNIT–5 : Applications (PYQ Solution)

AKTU · Artificial Intelligence (BCAI501) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Explain the concept of speech recognition. (Jan 2026)

Speech recognition (automatic speech recognition, ASR) is the technology that converts spoken language into text or commands. The audio is digitised, converted into features (such as MFCCs), and an acoustic model and language model together find the most likely word sequence. Modern systems use deep neural networks. Applications include voice assistants, dictation, call-centre analytics and captions. Accents, noise and homophones make it difficult.

Q2. Explain the concept of machine translation. (Jan 2025)

Machine translation (MT) is the automatic translation of text or speech from a source language to a target language by a computer. Approaches are rule-based (grammar and dictionaries), statistical (learned from parallel corpora) and neural MT (encoder-decoder and Transformer models), which is the present standard. Quality is measured with metrics such as BLEU. Examples are Google Translate and Bhashini. Ambiguity, idioms and low-resource languages remain challenges.

Q3. List various applications of Artificial Intelligence. (2023)

Main applications of AI are:

- **Healthcare: disease diagnosis, medical imaging, drug discovery.**
- **Finance: fraud detection, credit scoring, algorithmic trading.**
- **Education: adaptive learning and tutoring.**
- **Transport: self-driving cars, traffic control.**
- **Agriculture: crop monitoring and yield prediction.**
- **Language and speech: machine translation, voice assistants, chatbots.**
- **Vision: face recognition, surveillance, quality inspection.**

- **Robotics and manufacturing, gaming, e-commerce recommendation, cyber security.**

Q4. Discuss the concept of information retrieval. (2023)

Information retrieval (IR) is the process of finding relevant documents or information from a large collection in response to a user's query. Documents are indexed (for example with an inverted index), the query is matched using models such as Boolean, vector space (TF-IDF) or probabilistic models, and results are ranked by relevance. Search engines like Google are IR systems. Performance is measured by precision and recall.

Q5. Provide examples of real-world applications where information extraction is essential. (2024)

Information extraction (IE) automatically picks out structured facts (entities, relations, events) from unstructured text. It is essential in:

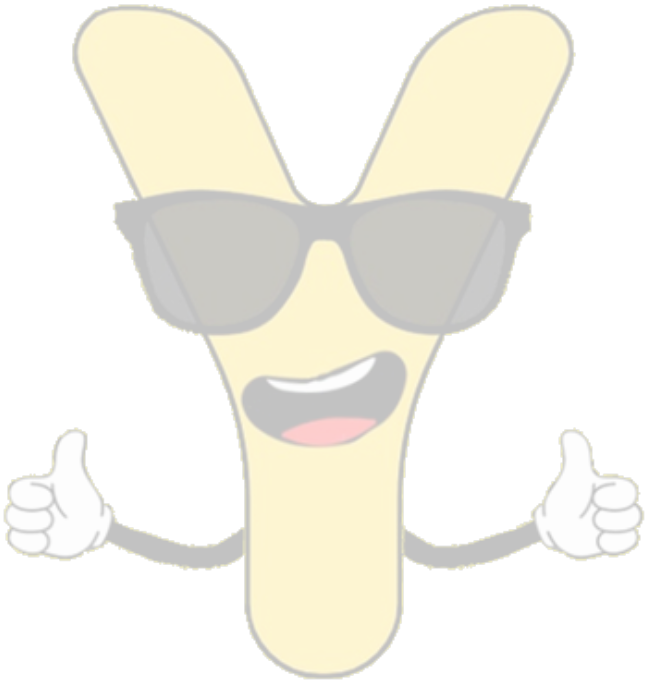
- **News and finance: extracting company names, deals and amounts from reports.**
- **Healthcare: pulling diseases, drugs and doses from clinical notes.**
- **Legal and HR: reading contracts and resumes to get names, skills and dates.**
- **E-commerce: product attributes and prices from web pages.**
- **Search engines and question answering: building knowledge graphs.**
- **Customer support: extracting issues from emails and reviews.**

Q6. Discuss the challenges associated with information retrieval in large and unstructured datasets. (2024)

Large unstructured data (text, images, audio, web pages) creates several challenges.

- **Scale: billions of documents need huge storage, efficient indexing and fast query response.**
- **Lack of structure: no fixed schema, so meaning has to be inferred.**
- **Ambiguity and vocabulary mismatch: synonyms and polysemy, e.g. "bank".**

- **Noise and quality: duplicate, spam, outdated and false content.**
- **Relevance ranking: understanding user intent and personalising results.**
- **Multilingual and multimedia content, and privacy and security issues.**



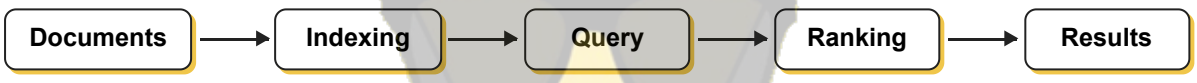
SECTION B Answers (≈250 Words Each)

Q1. Describe the role of information retrieval systems in modern AI applications, including search engines and recommendation systems. (Jan 2026, 2024)

Information retrieval (IR) systems find and rank relevant information from huge collections to satisfy a user's information need. In modern AI, IR is the **bridge between large data and intelligent applications**.

Basic pipeline

- **Crawling and collection** of documents.
- **Indexing:** an **inverted index** maps each term to the documents containing it.
- **Query processing:** cleaning, expansion, understanding intent.
- **Ranking:** TF-IDF, BM25, PageRank and learning-to-rank / neural rankers.
- **Evaluation:** precision, recall, F-measure, MAP, nDCG.



Information retrieval pipeline

Role in search engines

- Quickly return the most relevant pages out of billions in milliseconds.
- Use **semantic search** with embeddings to match meaning rather than exact words.
- Add personalisation, spelling correction, auto-complete and featured answers.

Role in recommendation systems

- Retrieve items (movies, products, songs) relevant to a user, using **content-based** matching, **collaborative filtering** (similar users) or hybrid models, then **rank** them.
- Used by Netflix, Amazon, YouTube and Spotify.

Other roles in modern AI

- **Retrieval-augmented generation (RAG):** chatbots retrieve documents and give them to a language model so answers are accurate and current.
- **Question answering, enterprise search, legal and medical search, digital libraries.**

- **Multimodal retrieval:** image, audio and video search.
- Feeds data for knowledge graphs and information extraction.

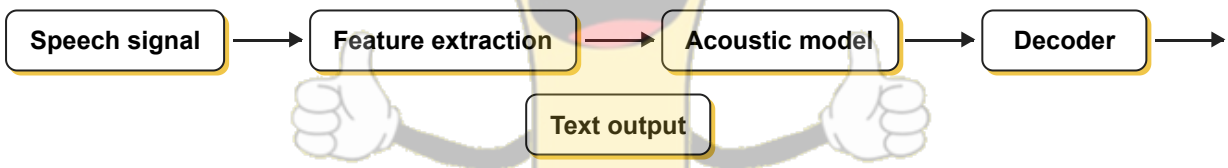
Application	IR technique
Web search	Inverted index + PageRank
Recommendation	Collaborative filtering, embeddings
Chatbots	RAG, vector search
Library search	Boolean / vector model

Challenges: scale, ambiguity, relevance, bias and filter bubbles, privacy.

Conclusion: IR provides the retrieval and ranking engine behind search, recommendations and modern LLM-based assistants.

Q2. Explain the architecture and functioning of speech recognition systems. (Jan 2025)

A **speech recognition system** converts a speech signal into the most probable word sequence. Statistically, it finds W that maximises $P(W \mid X) \propto P(X \mid W) \times P(W)$, where X is the acoustic observation.



Speech recognition architecture

Components

1. **Signal capture and pre-processing:** microphone input is digitised (sampling, 16 kHz), noise is reduced, and speech is split into short **frames** (about 25 ms).
2. **Feature extraction:** each frame is converted to features such as **MFCC** or filter-bank values representing the spectrum.
3. **Acoustic model:** gives P(X | phoneme/word), i.e. how likely a sound matches a phoneme. Classical models are **HMM-GMM**; modern models are **DNN, LSTM, CTC, Transformer (wav2vec, Whisper)**.
4. **Pronunciation lexicon:** maps words to phoneme sequences.
5. **Language model:** gives P(W), the likelihood of a word sequence (n-gram or neural), so that "recognise speech" is preferred over "wreck a nice beach".

6. **Decoder:** searches (Viterbi or beam search) for the best word sequence combining acoustic, lexicon and language scores.
7. **Post-processing:** punctuation, capitalisation and confidence scores.

Functioning: the user speaks; the audio is framed and converted to features; the acoustic model scores phonemes; the decoder combines these with the language model and outputs text.

Evaluation: Word Error Rate (WER) = (Substitutions + Deletions + Insertions) / words in reference.

Challenges: accents, noise, speaker variability, continuous speech without pauses, homophones, domain vocabulary.

Applications: voice assistants, dictation, captioning, voice search, call analytics.

Conclusion: speech recognition combines signal processing, acoustic and language models and a decoder to turn speech into text.

Q3. What is the role of the following in AI: (i) Machine Translation (ii) Speech Recognition. (2023)

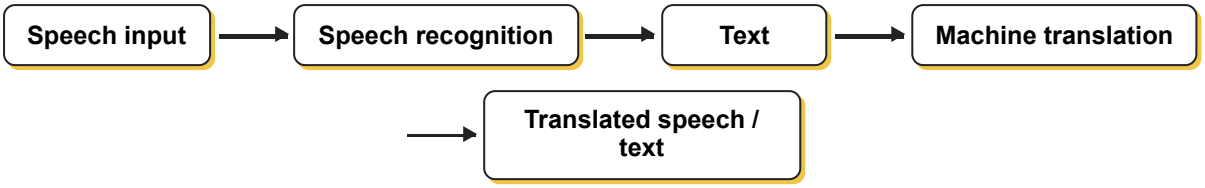
(i) Machine Translation (MT)

- MT automatically translates text or speech between languages, breaking the **language barrier**.
- Approaches: **rule-based**, **statistical** and **neural (Transformer-based)** MT.
- Role in AI: it is a core NLP task that tests language understanding and generation, and it drives research on **sequence-to-sequence models** and **attention**.
- Uses: real-time translation of web pages, chat and subtitles, cross-language search and customer support, government and education services in Indian languages.

(ii) Speech Recognition

- It converts spoken language into text using **acoustic models, language models and decoders**.
- Role in AI: it is the **input interface** for natural human-machine interaction through voice, which needs no keyboard.
- Uses: voice assistants (Alexa, Siri, Google Assistant), dictation, voice banking, hands-free car control, accessibility for visually or physically challenged users,

automatic captions and call centre analytics.



Speech-to-speech translation pipeline

Feature	Machine translation	Speech recognition
Input	Text in language A	Audio
Output	Text in language B	Text
Key models	Encoder-decoder, Transformer	Acoustic model + language model
Measure	BLEU	Word error rate

Combined use: speech recognition + MT + text-to-speech produces live voice translation.

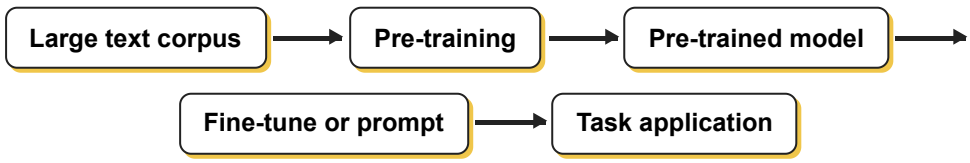
Conclusion: both technologies make AI systems more natural and accessible by handling human language and speech.

Q4. Explain the importance of pre-trained language models in various AI applications. (2024)

A **pre-trained language model (PLM)** is a neural model, usually a **Transformer**, trained on very large text data to learn grammar, meaning and world knowledge, and then **adapted (fine-tuned)** to specific tasks. Examples: **BERT, GPT, T5, RoBERTa, LLaMA**.

How they work

- **Pre-training:** self-supervised objectives such as **masked word prediction** (BERT) or **next-word prediction** (GPT) on billions of words.
- **Fine-tuning or prompting:** a small labelled data set (or just an instruction) adapts the model to a task.



Pre-train then adapt

Importance

- **Transfer learning:** knowledge learned once is reused, so tasks need much **less labelled data and training time**.
- **Better accuracy** on most NLP benchmarks.
- **Contextual understanding:** word meaning depends on context ("bank" of a river or money).
- **Versatility:** one model serves many tasks.
- **Few-shot and zero-shot ability** in large models.
- Reduces cost for small organisations since they can use open models.

Applications

- Machine translation, summarisation, **question answering**, **chatbots** and virtual assistants.
- **Sentiment analysis**, text classification, named entity recognition.
- **Search and recommendation** (semantic search, RAG).
- **Code generation**, content writing, speech and multimodal systems.

Limitations: huge computational cost, bias in training data, **hallucination**, privacy concerns and lack of transparency.

Model	Type	Typical use
BERT	Encoder	Classification, QA
GPT	Decoder	Generation, chat
T5	Encoder-decoder	Translation, summary

Conclusion: pre-trained language models have become the foundation of modern NLP because they give high accuracy with little task data.

SECTION C Answers (≈500 Words Each)

Q1. Explain planning algorithms in AI, such as STRIPS and hierarchical task networks. (Jan 2026)

Planning is the task of finding a **sequence of actions** that takes an agent from an **initial state** to a **goal state**. Unlike plain search, planning uses a **factored representation** of states and actions, which allows powerful heuristics and decomposition.

1. STRIPS (Stanford Research Institute Problem Solver)

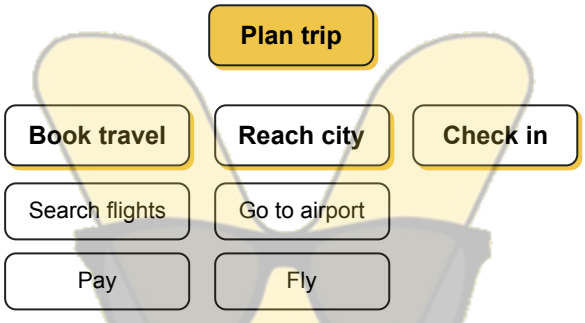
- A **state** is a set of true ground literals (facts), e.g. On(A, Table), Clear(A), HandEmpty.
- A **goal** is a set of literals to be made true.
- An **action** (operator) has three parts: **name and parameters**, **PRECONDITIONS** (must be true to apply) and **EFFECT**, divided into an **add list** (becomes true) and **delete list** (becomes false). Anything not mentioned is unchanged (solving the **frame problem** by assumption).
- **Example (blocks world):**

```
Action: Unstack(x, y)
  Precond: On(x, y), Clear(x), HandEmpty
  Delete:  On(x, y), Clear(x), HandEmpty
  Add:     Holding(x), Clear(y)
```

- **Planning as search:**
- **Forward (progression) search:** start at the initial state and apply applicable actions.
- **Backward (regression) search:** start from the goal and find actions that achieve part of it.
- **Partial-order planning:** actions are ordered only when necessary, using **causal links**.
- **GraphPlan / SATPlan / heuristics** (ignore delete lists) speed up planning.
- **Worked example:** Initial: On(A,B), On(B,Table), Clear(A), HandEmpty. Goal: Holding(A). Plan: Unstack(A, B). The precondition is satisfied; the result adds Holding(A) and Clear(B).

2. Hierarchical Task Network (HTN) planning

- Planning is done by **decomposing high-level tasks into subtasks** until only **primitive actions** remain.
- Components: **tasks** (primitive and compound), **methods** (ways to decompose a compound task, with preconditions) and a **task network** with ordering constraints.
- **Example:** Task *TravelToCity* has method 1 (by air): BookTicket, GoToAirport, Fly, GoToHotel; method 2 (by train): BookTrain, GoToStation, Ride, GoToHotel. The planner selects the method whose preconditions hold and expands each subtask further.
- **Advantages:** uses expert domain knowledge, so it searches much less; plans are natural for humans and scale to large problems (games, robotics, logistics). **Limit:** methods must be written by hand.



HTN decomposition of a task

Comparison

Feature	STRIPS planning	HTN planning
Basic idea	Search by actions to the goal	Decompose tasks into subtasks
Knowledge	Preconditions and effects only	Plus methods (recipes)
Search size	Large	Reduced
Goal form	Set of literals	Task network
Flexibility	General	Needs domain knowledge

Other issues: conditional and contingency planning for uncertainty, **replanning**, time and resources, and **PDDL** as a common planning language.

Conclusion: STRIPS gives a simple, general action model for planning as search, while HTN exploits hierarchical knowledge to plan complex tasks efficiently.

Brought to you By Ycotes : https://theycotes.com

Q2. Analyze the speech recognition techniques including acoustic modeling and language modeling. (Jan 2026)

Speech recognition chooses the word sequence W that maximises P(W | X) = P(X | W) × P(W) / P(X). Here P(X | W) comes from the **acoustic model** and P(W) from the **language model**; a **decoder** combines them.

Pre-processing and features

- The signal is sampled, divided into 20-30 ms frames and converted to **MFCC** (Mel-frequency cepstral coefficients), filter-bank or spectrogram features that mimic human hearing.

1. Acoustic modeling (relation between sound and phonemes)

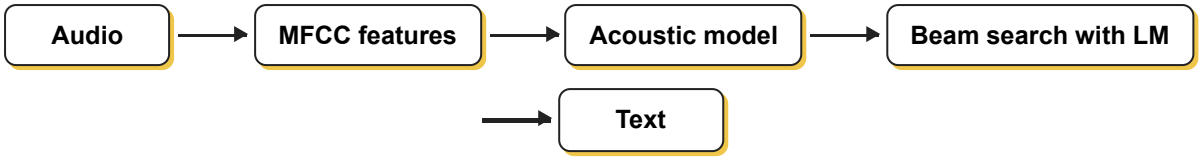
- **Template matching with Dynamic Time Warping (DTW):** compares the input with stored templates; used in early isolated-word systems.
- **HMM-GMM:** each phoneme is a **Hidden Markov Model** with states; Gaussian mixtures give the probability of features in each state; the **Viterbi algorithm** finds the best state sequence; the **Baum-Welch (EM) algorithm** trains it.
- **Hybrid DNN-HMM:** deep neural networks replace GMMs and give better state probabilities.
- **End-to-end models: CTC (Connectionist Temporal Classification)**, **RNN-Transducer**, **attention encoder-decoder** and **Transformer/Conformer** models (**wav2vec 2.0, Whisper**) learn directly from audio to characters or word pieces without separate lexicon.
- Context-dependent **triphones** capture co-articulation effects.

2. Language modeling (which word sequences are likely)

- **N-gram model:** P(w_n | w_(n-2), w_(n-1)); trained by counting, with **smoothing** (Laplace, Kneser-Ney, back-off) for unseen n-grams. Example: P("speech" | "recognize") is higher than P("beach" | "recognize").
- **Neural language models:** RNN/LSTM and **Transformer** models capture long-range context.
- **Perplexity** measures the quality: lower is better.

3. Decoding

- **Viterbi / beam search** over a graph (lexicon + language model) with scaled score = log P_acoustic + λ log P_LM. A **pronunciation lexicon** maps words to phonemes.



Speech recognition techniques

Comparison

Technique	Strength	Weakness
DTW	Simple	Small vocabulary
HMM-GMM	Mature, probabilistic	Needs hand-built parts
DNN-HMM	Higher accuracy	Complex pipeline
End-to-end (Whisper)	Simple, robust	Needs big data

Evaluation: Word Error Rate (WER) = (S + D + I) / N. Example: reference of 10 words with 1 substitution, 1 deletion, 1 insertion gives WER = 3/10 = 30 percent.

Challenges: noise, accents, speaker variation, overlapping speakers, code-mixed speech (Hinglish), low-resource languages.

Conclusion: acoustic models handle what was heard and language models what is plausible; modern end-to-end neural systems combine both with high accuracy.

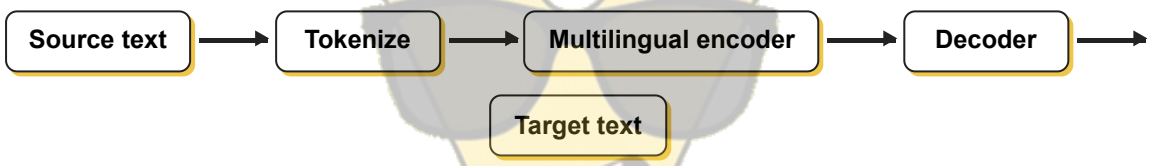
Q3. Discuss the challenges of multilingual support in machine translation systems. (Jan 2025)

Multilingual machine translation aims to translate between many languages, often with one model. It increases reach but creates many difficulties.

Main challenges

- **Low-resource languages:** very little **parallel data** exists for many Indian and African languages, so quality is poor. Remedies: back-translation, transfer learning, multilingual pre-training.
- **Linguistic diversity:** different **word order** (English SVO, Hindi SOV), **morphology** (agglutinative languages like Tamil and Turkish create long words), gender, honorifics and case systems.
- **Different scripts:** Devanagari, Arabic, Chinese need **tokenization** and **transliteration** handling; scripts without spaces (Chinese, Thai) are hard to segment.

- **Ambiguity:** words with several meanings need context. "bank" can be river or money.
- **Idioms, culture and humour:** literal translation of idioms gives wrong output.
- **Code-mixing:** users mix languages (Hinglish) within a sentence.
- **Named entities and transliteration** of names, places.
- **Domain and dialect variations:** medical, legal text; regional dialects.
- **Negative interference and capacity:** in a single multilingual model, high-resource languages dominate and low-resource ones suffer, the **curse of multilinguality**.
- **Zero-shot translation** between language pairs not seen in training is less reliable.
- **Evaluation:** BLEU is weak for morphologically rich languages; human evaluation is costly.
- **Bias and ethics:** gender bias (translating "doctor" as male), harmful or sensitive content.
- **Computational cost** of large shared vocabularies and models.



Multilingual NMT pipeline

Solutions

- **Shared sub-word vocabularies** (BPE, SentencePiece), language tags, **multilingual Transformers** (mBART, M2M-100, NLLB, IndicTrans).
- **Data augmentation**, back-translation, pivoting via English.
- **Language-specific adapters** and mixtures of experts.
- Human-in-the-loop evaluation and quality estimation.

Challenge	Example	Remedy
Low resource	Maithili	Transfer learning
Word order	English to Hindi	Attention models
Morphology	Tamil	Sub-word units
Code mixing	Hinglish	Mixed training data

Conclusion: multilingual MT must deal with data scarcity, linguistic differences and cultural meaning; shared multilingual neural models and better data help, but quality

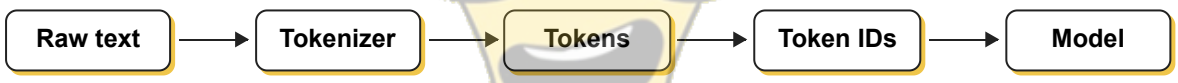
still varies strongly across languages.

Q4. Explain how tokenization impacts natural language processing tasks. (Jan 2025)

Tokenization is the process of splitting raw text into smaller units called **tokens** (words, sub-words, characters or punctuation) that a model can process. It is the **first step of every NLP pipeline**, and its design affects accuracy, speed and vocabulary size.

Types of tokenization

- **Word-level:** split on spaces and punctuation. "I love AI" gives [I, love, AI]. Simple, but gives a huge vocabulary and **out-of-vocabulary (OOV)** words.
- **Character-level:** every character is a token. No OOV problem, but sequences are long and meaning is weak.
- **Sub-word level: BPE (Byte-Pair Encoding), WordPiece, SentencePiece, Unigram** split rare words into pieces: "unhappiness" gives [un, happi, ness]. This balances vocabulary size and coverage; used in BERT and GPT.
- **Sentence tokenization** splits a document into sentences.



Tokenization in the NLP pipeline

Impact on NLP tasks

- **Vocabulary size and memory:** word-level needs a huge embedding table; sub-word keeps it manageable (30-50 thousand tokens).
- **OOV handling:** sub-words let a model represent unseen or misspelt words.
- **Model accuracy:** poor tokenization breaks meaning; for example splitting "New York" or "don't" wrongly harms **named entity recognition and sentiment analysis**.
- **Morphologically rich and unsegmented languages:** Hindi, Tamil, Turkish and Chinese need language-aware tokenizers; space-based splitting fails for Chinese and Japanese.
- **Sequence length and cost:** more tokens mean more computation; transformer cost grows with sequence length, and models have **context-window limits**. Languages that tokenize poorly (many tokens per word) are slower and more expensive.

- **Machine translation and text generation:** sub-word units help generate rare words and names.
- **Information retrieval:** tokens form the index terms; combined with **stemming, lemmatization and stop-word removal** they affect recall and precision.
- **Fairness:** tokenizers trained mainly on English split other languages into many tokens, reducing quality.

Example: "Don't stop!" word tokenizer: [Do, n't, stop, !]; a simple space split: [Don't, stop!]. Different tokens lead to different embeddings and results.

Method	Vocabulary	OOV	Used in
Word	Very large	Yes	Classical NLP
Character	Tiny	No	Some spelling models
Sub-word (BPE)	Medium	Rare	GPT, BERT, T5

Conclusion: tokenization decides what the model sees as the basic unit, so a good tokenizer improves accuracy, efficiency and language coverage in all NLP tasks.

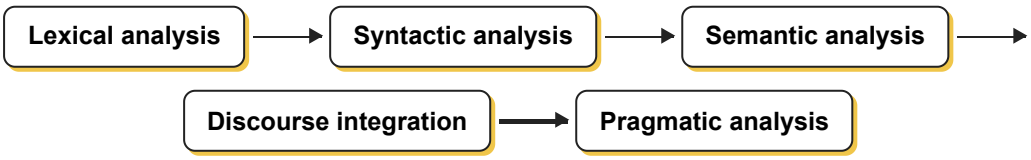
Q5. Discuss the role of NLP in AI. Describe the stages of natural language processing in artificial intelligence. (2023)

Natural Language Processing (NLP) is the branch of AI that enables computers to **understand, interpret and generate human language**. It includes **Natural Language Understanding (NLU)** and **Natural Language Generation (NLG)**.

Role of NLP in AI

- Provides **natural human-computer interaction** by text and voice.
- Applications: **machine translation, chatbots and virtual assistants, speech recognition, sentiment analysis, question answering, summarisation, information retrieval and extraction, spam filtering, grammar checking**.
- Allows machines to learn from the large volume of unstructured text.
- Forms the base of large language models.

Stages of NLP



Stages of natural language processing

1. **Lexical analysis (morphological):** the text is divided into paragraphs, sentences and words (**tokens**); words are reduced by stemming or lemmatisation and tagged with parts of speech.
2. **Syntactic analysis (parsing):** checks the **grammar** and arranges words to show relationships, producing a **parse tree**. "The school goes to boy" is rejected as ungrammatical.
3. **Semantic analysis:** finds the **literal meaning** of the sentence and removes meaningless forms. "Hot ice-cream" is semantically odd. It also handles **word-sense disambiguation**.
4. **Discourse integration:** the meaning of a sentence depends on the **previous and following sentences**; resolves references such as "he", "it" (anaphora).
5. **Pragmatic analysis:** interprets the **intended meaning in context**, using real-world knowledge. "Can you pass the salt?" is a request, not a question about ability.

Example: "Ravi ate the apple. He liked it." Lexical splits tokens; syntax parses each sentence; semantics links "ate" to the actor and object; discourse links "He" to Ravi and "it" to the apple; pragmatics understands Ravi enjoyed the fruit.

Stage	Question answered	Output
Lexical	What are the words?	Tokens, tags
Syntactic	Is it grammatical?	Parse tree
Semantic	What does it mean?	Logical form
Discourse	How do sentences connect?	Resolved references
Pragmatic	What is intended?	Real meaning

Challenges: ambiguity, idioms, context, multiple languages.

Conclusion: NLP lets machines work with human language, and it proceeds through lexical, syntactic, semantic, discourse and pragmatic stages.

Q6. What is Robotics? Differentiate between a robotic system and other AI programs. Describe the various components of a robot. How does computer vision contribute in robotics? (2023)

Robotics is the branch of engineering and AI that designs, builds and programs **robots**, machines that sense their environment, make decisions and perform

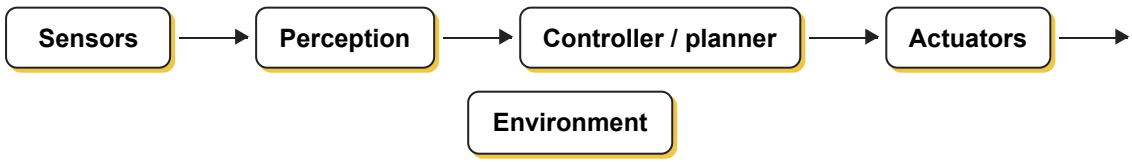
physical actions. An **AI robot** is an agent with sensors and effectors that works in the real world.

Robotic system vs other AI programs

Basis	Robotic system	Other AI program
Form	Physical body in real world	Software only
Environment	Real, noisy, dynamic	Digital or simulated
Input	Sensors (camera, lidar, touch)	Data, text, files
Output	Physical actions (motion)	Information or decision
Issues	Sensor error, motion uncertainty, safety, real-time control	Data quality, accuracy
Example	Warehouse robot, drone	Chess program, spam filter

Components of a robot

- **Manipulator / mechanical structure:** links and joints, the body, arm, legs or wheels.
- **End effector:** gripper, hand or tool at the end of an arm.
- **Actuators:** electric motors, servos, hydraulic or pneumatic cylinders that produce motion.
- **Sensors:** cameras, ultrasonic, infrared, lidar, touch, force, gyroscope and GPS for perception.
- **Controller (brain):** microcontroller or computer running the control and AI software; does planning and decision making.
- **Power supply:** battery or mains.
- **Drive/transmission system:** gears and belts.
- **Software and communication interfaces.**



Robot control loop

Role of computer vision in robotics

- **Object recognition and detection:** identifies parts, people and obstacles using CNNs.

- **Localisation and mapping (visual SLAM):** the robot finds where it is and builds a map.
- **Navigation and obstacle avoidance:** detects free space, lanes, signs.
- **Visual servoing and manipulation:** guides the arm to grasp using camera feedback and **pose estimation**.
- **Inspection and quality control** in manufacturing, defect detection.
- **Human-robot interaction:** face, gesture and emotion recognition.
- **3D perception:** depth from stereo cameras or depth sensors.
- Applications: self-driving cars, surgical robots, pick-and-place arms, agricultural robots, drones.

Conclusion: robotics integrates sensing, intelligence and action in a physical body, and computer vision gives the robot its main sense to understand and interact with the world.

Q7. What are language models, and how do they contribute to natural language processing tasks? (2024)

A **language model (LM)** is a model that assigns a **probability to a sequence of words**, P(w1, w2, ..., wn), and so can predict the **next word** given the previous words: P(w_n | w_1 ... w_(n-1)). It captures the grammar and meaning patterns of a language.

Types

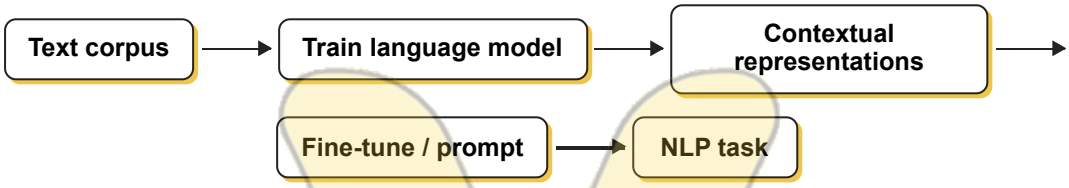
- **Statistical (n-gram) models:** use counts of word sequences; P(w_n | w_(n-1)) = count(w_(n-1), w_n) / count(w_(n-1)). Simple, but limited context and data sparsity (needs smoothing).
- **Neural LMs:** feed-forward, **RNN/LSTM** models learn **word embeddings** and longer context.
- **Transformer-based LMs:** use **self-attention**; examples are **BERT** (masked LM, encoder), **GPT** (autoregressive, decoder), **T5** (encoder-decoder). Large ones are called **large language models (LLMs)**.

Example (bigram). With counts "I love" = 3 and "I" = 4, P(love | I) = 3/4 = 0.75.

Evaluation: perplexity (lower is better), plus task benchmarks.

Contribution to NLP tasks

- **Speech recognition:** picks the likely word sequence among similar-sounding options.
- **Machine translation:** judges the fluency of output sentences.
- **Text generation, summarisation and dialogue:** generate coherent text by predicting words one after another.
- **Question answering and information retrieval:** understand questions and passages; semantic search.
- **Sentiment analysis, classification, named entity recognition:** contextual embeddings from a pre-trained LM give high accuracy.
- **Spelling and grammar correction, predictive typing and auto-complete.**
- **Code generation and reasoning** in large models.



Language models in NLP

Model	Context	Typical task
n-gram	1-4 words	Speech decoding, typing
LSTM	Long sequences	Translation (older)
BERT	Both sides	Classification, QA
GPT	Left to right	Generation, chat

Limits: may produce false but fluent text (hallucination), reflect training bias, and need large computing power.

Conclusion: language models give machines a statistical understanding of language, which is the foundation of almost all modern NLP applications.
