Natural Language Processing (TYAIML)

Unit-1

Origin of NLP

Natural Language Processing (NLP) has its roots in several disciplines, including linguistics, computer science, artificial intelligence, and cognitive psychology.

Here’s a brief overview of its origin and evolution:

1. **Alan Turing’s Influence (1950)**
   - Turing proposed the idea of machine intelligence in his paper *"Computing Machinery and Intelligence"*, introducing the **Turing Test**—a measure of a machine's ability to exhibit human-like conversation.
2. **The Birth of Machine Translation (1950s)**
   - The Georgetown-IBM Experiment (1954) demonstrated the first automated Russian-to-English translation system, sparking interest in computational linguistics.
3. **Noam Chomsky’s Syntactic Structures (1957)**
   - Chomsky’s work on **formal grammars** influenced early NLP systems, leading to rule-based parsing techniques.
4. **ELIZA (1966) & SHRDLU (1972)**
   - **ELIZA**, created by Joseph Weizenbaum, simulated a Rogerian psychotherapist using pattern matching.
   - **SHRDLU**, by Terry Winograd, demonstrated a limited but intelligent NLP system for a blocks-world environment.
5. **Shift to Probabilistic Models (1990s)**
   - Researchers moved from rigid rule-based systems to **statistical models**, leveraging large text corpora.
   - Hidden Markov Models (HMMs) and **n-gram language models** became popular for speech recognition and machine translation.
6. **IBM’s Statistical Machine Translation (1990s)**
   - IBM’s **Candide** project used statistical methods, improving translation accuracy over rule-based systems.
7. **Neural Networks & Word Embeddings**
   - **Word2Vec (2013)** by Google introduced dense vector representations of words, capturing semantic relationships.

   - **Sequence-to-Sequence Models (2014)** enabled better machine translation (e.g., Google Translate).

8. **Transformer Revolution (2017–Present)**
   - **Attention Mechanisms & Transformers** (introduced in *"Attention Is All You Need"*) led to breakthroughs like **BERT, GPT, and T5**.
   - Large Language Models (LLMs) like **ChatGPT (2022)** and **GPT-4 (2023)** demonstrated human-like text generation.

History of NLP

The **history of Natural Language Processing (NLP)** spans several decades, evolving from rule-based systems to modern deep learning approaches.

Here's a detailed overview of its key milestones:

1. Foundations (1940s–1950s)

- **1940s–1950s:** Early computational linguistics emerged with **Alan Turing's** work on machine intelligence and the **Turing Test** (1950), which proposed evaluating a machine's ability to exhibit human-like conversation.
- **1954: Georgetown-IBM Experiment** – One of the first NLP demonstrations, where a machine translated over 60 Russian sentences into English using rule-based methods.

2. Rule-Based Systems (1960s–1980s)

- **1960s: ELIZA** (1966), an early chatbot by Joseph Weizenbaum, simulated conversation using pattern matching but had no real understanding.
- **1970s: SHRDLU** (1972) by Terry Winograd demonstrated limited natural language understanding in a virtual blocks world.
- **1980s:** Rise of **expert systems** and hand-crafted grammars (e.g., **Chomsky's transformational grammar**). Systems relied on **symbolic AI**, with limited scalability.

3. Statistical NLP & Machine Learning (1990s–2000s)

- **1990s:** Shift from rule-based to **statistical methods** due to increased computational power and data availability.
  - **Hidden Markov Models (HMMs)** and **n-gram models** improved speech recognition and machine translation.
  - **IBM's statistical machine translation** (e.g., **BLEU metric**, 2002) replaced earlier rule-based approaches.
- **2000s:** Introduction of **machine learning algorithms** (e.g., **SVMs, CRFs**) for tasks like part-of-speech tagging and named entity recognition.

4. Rise of Deep Learning (2010s–Present)

- **2013: Word embeddings** (e.g., **Word2Vec** by Mikolov) enabled machines to capture semantic word relationships.
- **2015–2017: Sequence-to-sequence models** (Seq2Seq) and **attention mechanisms** improved machine translation (e.g., **Google's Neural Machine Translation**).
- **2018: Transformer architecture** (Vaswani et al.) revolutionized NLP with models like **BERT, GPT**, and **T5**, enabling **contextual understanding**.
- **2020s:** Large-scale **pre-trained language models** (e.g., **GPT-3, GPT-4, PaLM, LLaMA**) dominate NLP, achieving human-like text generation and comprehension.

Key Applications over Time

- **Machine Translation** (e.g., Google Translate)
- **Speech Recognition** (e.g., Siri, Alexa)
- **Sentiment Analysis & Chatbots** (e.g., ChatGPT)
- **Text Summarization & Question Answering** (e.g., BERT-based systems)

Future Directions

- **Multimodal NLP** (combining text, images, and audio)
- **Ethical AI & Bias Mitigation**
- **Few-shot & Zero-shot Learning** (reducing reliance on massive datasets)

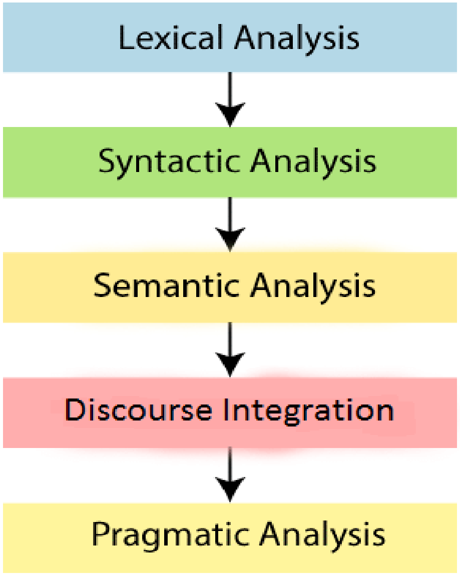
NLP has evolved from simple rule-based systems to sophisticated AI models, transforming how humans interact with machines. The field continues to advance rapidly with new breakthroughs in **large language models (LLMs)** and **generative AI**.

Stages in NLP (Hierarchical)

This structured approach mirrors how humans process language from recognizing words to grasping deeper context.

Why These Stages Matter:

- **Hierarchy**: Each stage builds on the previous one, from raw text to nuanced understanding.
- **Applications**: Used in chatbots, translation, voice assistants, and more.
- **Modern NLP**: Deep learning models (e.g., BERT, GPT) often handle these stages implicitly in an end-to-end manner.



The figure outlines the **key stages (phases/levels) in Natural Language Processing (NLP)**, which represent the hierarchical process of understanding and interpreting human language computationally. Here's a breakdown of each stage:

1. **Lexical Analysis**

- **Purpose**: Deals with the structure of words and their basic meanings.
- **Tasks**:
  - **Tokenization**: Splitting text into words, symbols, or phrases (tokens).
  - **Morphological Analysis**: Breaking words into morphemes (e.g., "unhappiness" → "un" + "happy" + "ness").
  - **Stopword Removal**: Filtering out common but insignificant words (e.g., "the," "and").
- **Example**:
  - Input: "Cats chase mice."
  - Output: Tokens ["Cats", "chase", "mice"].

2. **Syntactic Analysis (Parsing)**

- **Purpose**: Examines sentence structure and grammar.
- **Tasks**:
  - **Part-of-Speech (POS) Tagging**: Labelling words as nouns, verbs, etc.
  - **Parsing**: Building parse trees to show grammatical relationships (e.g., subject-verb-object).
- **Example**:
  - Input: "Cats chase mice."
  - Parse Tree: text

```
(S (NP Cats) (VP (V chase) (NP mice)))
```

3. **Semantic Analysis**

- **Purpose**: Extracts the literal meaning of words and sentences.
- **Tasks**:
  - **Word Sense Disambiguation**: Resolving multiple meanings (e.g., "bank" as financial institution vs. river edge).
  - **Semantic Role Labeling**: Identifying "who did what to whom" (e.g., "Cats" = agent, "mice" = patient).
- **Example**:
  - Output: "Cats (agent) chase (action) mice (patient)."

4. **Discourse Integration**

- **Purpose**: Connects sentences to understand broader context.
- **Tasks**:
  - **Coreference Resolution**: Linking pronouns to nouns (e.g., "He" refers to "John").
  - **Cohesion Analysis**: Ensuring logical flow between sentences.
- **Example**:
  - Input: "John saw a cat. It was black."
  - Output: "It" → "cat".

5. **Pragmatic Analysis**

- **Purpose**: Interprets meaning in real-world context (intent, tone, implied meaning).
- **Tasks**:
  - **Sentiment Analysis**: Detecting emotions (e.g., sarcasm, anger).
  - **Intent Recognition**: Understanding user goals (e.g., a question vs. a command).
- **Example**:
  - Input: "Great, another rainy day!"
  - Output: Sarcasm (negative sentiment).

Stages in NLP

Natural Language Processing (NLP) involves several stages to transform raw text into meaningful and actionable information.

Here are the key stages of NLP:

1. **Text Preprocessing**

- **Tokenization**: Splitting text into words, phrases, or sentences (tokens).
- **Normalization**: Converting text to a standard form (e.g., lowercase, removing accents).

- **Stopword Removal**: Eliminating common but insignificant words (e.g., "the," "is").
- **Stemming & Lemmatization**: Reducing words to their base/root form (e.g., "running" → "run").
- **Noise Removal**: Cleaning text by removing irrelevant characters (e.g., HTML tags, punctuation).

2. **Text Representation**
   - **Bag of Words (BoW): Representing text as word frequencies.**
   - **TF-IDF (Term Frequency-Inverse Document Frequency): Weighing words based on importance.**
   - **Word Embeddings (e.g., Word2Vec, GloVe): Capturing semantic meaning via dense vectors.**
   - **Contextual Embeddings (e.g., BERT, GPT): Using deep learning models for context-aware representations.**
3. **Syntactic Analysis (Understanding Structure)**
   - **Part-of-Speech (POS) Tagging: Labelling words with grammatical roles (e.g., noun, verb).**
   - **Parsing (Dependency & Constituency): Analysing sentence structure and relationships.**
   - **Named Entity Recognition (NER): Identifying entities (e.g., people, organizations).**
4. **Semantic Analysis (Understanding Meaning)**
   - **Word Sense Disambiguation**: Determining the correct meaning of words in context.
   - **Semantic Role Labelling (SRL)**: Identifying predicate-argument structures.
   - **Sentiment Analysis**: Detecting emotions or opinions in text.
5. **Pragmatic Analysis (Contextual Understanding)**
   - **Coreference Resolution**: Linking pronouns to their referents (e.g., "he" → "John").
   - **Discourse Analysis**: Understanding how sentences connect to form coherent meaning.
   - **Intent Recognition (in Dialogue Systems)**: Identifying user goals in conversations.
6. **Application-Specific Tasks**
   - **Machine Translation**: Converting text from one language to another.
   - **Text Summarization**: Generating concise summaries of long documents.
   - **Question Answering (QA)**: Extracting answers from text based on queries.
   - **Chatbots & Virtual Assistants**: Simulating human-like conversations.
   - **Text Classification**: Categorizing text (e.g., spam detection, topic labeling).
7. **Evaluation & Optimization**

- **Model Training & Fine-Tuning**: Using machine/deep learning to improve performance.
- **Performance Metrics**: Evaluating accuracy, precision, recall, F1-score, BLEU (for translation), etc.
- **Human Evaluation**: Assessing real-world usability and coherence.

These stages may vary based on the specific NLP task (e.g., sentiment analysis vs. machine translation), but they provide a general pipeline for processing and understanding natural language. Modern NLP heavily relies on **deep learning** (transformers, LLMs) to handle many of these stages end-to-end.

Ambiguities in Natural Language Processing (NLP)

Ambiguity occurs when a word, phrase, or sentence has multiple possible interpretations. It poses challenges in NLP tasks like machine translation, speech recognition, and sentiment analysis. Ambiguities exist in **English** as well as **Indian regional languages** (e.g., Hindi, Bengali, Tamil, Telugu, etc.), though their manifestations differ due to linguistic structures.

1. Types of Ambiguities in English-

(A) Lexical Ambiguity (Word-Level Ambiguity)

- A single word has multiple meanings.
- **Examples**:
  - "Bank" → Financial institution / Riverbank.
  - "Bat" → Animal / Sports equipment.
  - "Light" → Not heavy / Illumination.

(B) Syntactic Ambiguity (Structural Ambiguity)

- A sentence has multiple possible grammatical structures.
- **Examples**:
  - "I saw the man with the telescope."
    - Did I use the telescope to see the man?
    - Or did the man have a telescope?
  - "Visiting relatives can be boring."
    - Are the relatives visiting?
    - Or is someone visiting relatives?

(C) Semantic Ambiguity (Meaning-Based Ambiguity)

- A phrase or sentence has multiple interpretations due to word relationships.
- **Examples**:

    - "The chicken is ready to eat."
        - Is the chicken cooked and ready to be eaten?
        - Or is the chicken hungry and ready to eat food?
    - "He fed her cat food."
        - Did he feed her cat some food?
        - Or did he feed her food meant for cats?

(D) Pragmatic Ambiguity (Context-Based Ambiguity)

- The meaning changes based on real-world context or speaker's intent.
- **Examples**:
    - "It's cold in here."
        - Literal meaning: The temperature is low.
        - Pragmatic meaning: A request to close the window.
    - "Nice job!"
        - Could be genuine praise or sarcasm.

2. Types of Ambiguities in Indian Regional Languages-

Indian languages (e.g., Hindi, Bengali, Tamil, Telugu, etc.) have additional complexities due to:

- **Morphological richness** (more inflections than English).
- **Free word order** (subject-object-verb flexibility).
- **Homophones** (words that sound the same but differ in meaning).

(A) Lexical Ambiguity

- Many words have multiple meanings.
- **Examples (Hindi)**:
    - "काल" (Kaal) → Time / Death / Yama (god of death).
    - "पत्र" (Patra) → Letter (mail) / Leaf.

(B) Syntactic Ambiguity (Due to Free Word Order)

- Indian languages allow flexible sentence structures, leading to multiple interpretations.
- **Examples (Hindi)**:
    - "राम ने सीता को गीता दी।"
        - Did Ram give Sita to Geeta?
        - Or did Ram give Geeta (a book) to Sita?

(C) Morphological Ambiguity (Due to Suffixes & Sandhi)

- Words change meaning based on suffixes or compound formations.

- **Examples (Sanskrit-influenced languages)**:
    - "राजदर्शन" (Raja-darshan) → King’s visit / Seeing the king.

(D) Anaphora & Pronoun Ambiguity

- Pronouns may refer to multiple entities.
- **Examples (Hindi)**:
    - "राम ने श्याम से कहा कि वह जा रहा है।"
        - "He" (वह) could be Ram or Shyam.

(E) Homophonic Ambiguity (Same Pronunciation, Different Meaning)

- Common in Indian languages due to script variations.
- **Examples (Hindi)**:
    - "सोना" (Sona) → Gold / Sleep (verb).

3. Challenges in Resolving Ambiguities in Indian Languages-

1. **Lack of Standardization**: Many dialects and informal usages.
2. **Script Variations**: Same word written differently in different scripts (Devanagari vs. Urdu).
3. **Context Dependence**: Heavy reliance on context for disambiguation.
4. **Machine Translation Issues**: Direct translation often fails due to structural differences.

How NLP Handles Ambiguities?

- **Word Sense Disambiguation (WSD)** → Uses context to pick the right meaning.
- **Coreference Resolution** → Links pronouns to correct nouns.
- **Statistical & Neural Models (BERT, GPT)** → Learn from large datasets to infer meaning.

Applications of NLP

Following are the key applications of Natural Language Processing (NLP).

1. Machine Translation:

- Machine Translation (MT) is a subfield of Natural Language Processing (NLP)
- Focuses on the development of automated systems which are capable of translating text or speech from one language to another.
- The primary goal of machine translation is to facilitate communication between people who speak different languages.

Key Aspects of Machine Translation:

A. Rule-Based Machine Translation (RBMT):

- Based on linguistic rules and dictionaries.
- These systems used predefined grammatical and syntactical rules to translate text.

B. Statistical Machine Translation (SMT):

- Based on statistical models and large bilingual corpora (collection of authentic text or audio organized into datasets).
- These models learn patterns and probabilities from the data to make translation decisions.

C. Neural Machine Translation (NMT):

- Recent and advanced approach to machine translation.
- It employs artificial neural networks, specifically recurrent neural networks (RNNs) or transformers, to learn complex patterns and relationships in language data.
- NMT has shown significant improvements in translation quality compared to previous methods.

D. Online Translation Services:

- Popular online platforms such as Google Translate, Microsoft Translator, and DeepL use machine translation to provide instant translation services for users.

E. Multilingual and Zero-Shot Translation:

- Some advanced NMT models can handle multiple languages simultaneously, allowing for more efficient translation across language pairs.
- Zero-shot translation involves translating between language pairs that were not clearly part of the training data.

F. Domain-Specific Translation:

- MT systems can be design for specific domains, such as medical, legal, or technical translation.
- This customization enhances the accuracy and relevance of translations within specialized fields.

G. Post-Editing and Human-In-The-Loop:

- In professional settings, machine-generated translations are often reviewed and edited by human translators to ensure accuracy and quality.
- This collaboration between machines and humans is known as human-in-the-loop translation.

2. Information Retrieval-

- It involves the efficient and effective retrieval of information from large collections or databases.

- NLP plays a crucial role in information retrieval by enabling systems to understand, analyze, and extract relevant information from natural language data.

Key Aspects of Information Retrieval:

A. Search Engines

- NLP is used in search engines to understand user queries and retrieve relevant documents or web pages. It involves techniques like query parsing and semantic analysis.

B. Semantic Search

- Searching goes beyond traditional keyword matching. Semantic search systems understand the context of the user's query, providing more accurate and contextually relevant results.

3. Question Answering System-

- It aims to develop automated systems that are capable of understanding user queries in natural language and providing relevant and accurate answers.

Key Aspects of Question Answering System:

A. Understanding Natural Language Questions

- QA systems use NLP techniques to understand and interpret user questions, which can be modelled in various ways and may involve complex language structures.

B. Semantic Analysis:

- NLP helps in the semantic analysis of questions, allowing the system to grasp the meaning and context behind user queries.

C. Real-time Interaction

- Enabling real-time interaction where users can ask questions and receive immediate responses, enhancing the user experience.

4. Sentiment analysis-

- It involves determining the sentiment expressed in a piece of text.
- The primary goal is to identify and understand the emotions, opinions, and attitudes expressed by individuals toward a particular subject or topic.

Key Aspects of Sentiment analysis:

A. Sentiment Classification

- NLP techniques are used to classify text into different sentiment categories, such as positive, negative, or neutral.
- Machine learning models, including classifiers, are often employed for this purpose.

B. Social Media Monitoring

- Sentiment analysis is widely used to monitor and analyze opinions and sentiments expressed on social media platforms.
- Companies may use it to measure public perception of their brand, products, or services.

C. Customer Feedback Analysis

- Sentiment analysis helps businesses analyze customer reviews, feedback, and comments to understand customer satisfaction levels and identify areas for improvement.

D. Product Reviews

- E-commerce platforms leverage sentiment analysis to assess product reviews and ratings, helping potential buyers make informed decisions.

E. Brand Monitoring

- Companies use sentiment analysis to monitor online conversations and news articles related to their brand, helping them manage their online reputation.

F. Political Analysis

- Sentiment analysis is used in political campaigns to measure public sentiment towards candidates, policies, or political events.

G. Hate Speech Detection

- Sentiment analysis can be extended to identify and moderate hate speech or toxic comments on online platforms.

5. Text Categorization-

- Text Categorization, also known as text classification is a fundamental application of Natural Language Processing (NLP) that involves automatically assigning predefined categories or labels to text documents based on their content.

Key aspects of Text Categorization:

A. Document Classification

- Text categorization is widely used to classify documents into predefined categories, making it easier to organize and retrieve information.

B. Spam Detection

- In email filtering systems, text categorization is applied to differentiate between spam and genuine emails, helping to keep inboxes clean.

C. Topic Classification

- Identifying the main topics or themes in a document and categorizing it accordingly. This is useful in organizing and indexing large collections of documents.

D. News Categorization

- Categorizing news articles into topics such as politics, sports, technology, and entertainment to facilitate easy navigation and content recommendation.

6. Text Summarization-

- It involves the process of creating brief and clear summaries of lengthy documents while retaining the essential information.
- This application aims to capture the main points and key details within a document, enabling users to grasp the content quickly without reading the entire text.

Key aspects of Text Summarization:

A. Extractive Summarization

- Involves selecting and extracting important sentences or phrases directly from the original text to construct the summary.

B. Abstractive Summarization

- Involves generating new sentences that convey the essential information in a more reduced form.
- This approach often requires a deeper understanding of the content.

C. Document Summarization

- Summarizing entire documents, such as research papers, articles, or reports, to provide a brief overview of the main findings or key points.

D. News Summarization

- Automatically generating concise summaries of news articles to provide readers with a quick understanding of the main events and key information.

7. Named Entity Recognition (NER)-

- It involves identifying and classifying entities, such as names of people, organizations, locations, dates, and other specific types of information, within a given text.
- NER plays a fundamental role in extracting structured information from unstructured text data.

Key aspects of Named Entity Recognition:

A. Information Extraction

- Extract key information from text, enabling the identification and categorization of entities, which can be further used for various applications.

B. Named Entity Categorization

- Classifying identified entities into predefined categories, such as person names, company names, geographical locations, and more.

C. Healthcare Records Analysis:

- Used to identify and categorize entities within medical records, such as patient names, medical conditions, medications, and treatment dates.

D. Chatbots and Virtual Assistants:

- Improving the understanding and context of user queries by identifying entities and providing more accurate and relevant responses.

E. E-commerce Product Catalogs

- Identifying and categorizing entities within product descriptions, helping organize and enhance e-commerce product catalogs.