Chapter 1
NLP Overview

Er. Shiva Ram Dam

Assistant Professor

Gandaki University

Content:

1. Introduction to NLP
2. Origins and importance of NLP
3. Components of NLP
4. Phases in NLP
5. Challenges in NLP
6. Applications of NLP
7. NLP libraries: NLTK, Spacy, Gensim, Stanford CoreNLP

Introduction to NLP

Goal: Deep Understanding

- Requires context, linguistic structure, meanings…

Reality: Shallow Matching

- Requires robustness and scale
- Amazing successes, but fundamental limitations

What do computers understand?

- **Only Binary (0s and 1s)**
  - At the lowest level, computers understand **electric signals**:
    - **1 = ON (electricity flowing)**
    - **0 = OFF (no electricity)**
- Everything—text, numbers, images, videos, AI—ultimately becomes **patterns of 0s and 1s** inside the computer.
- They use Logic, Not Meaning:
  - If you show a computer the word **"apple"**, it does **not** know it is a fruit. It only sees:
    - 01100001 01110000 01110000 01101100 01100101
- Understanding meaning requires **AI models**, but even AI does not "understand" like humans—only predicts patterns.

They process specific types of data

- Computers “understand” data in these forms:

Data Type	Example	Computer’s Interpretation
Numbers	15, 3.14	Binary form of numbers
Text	"Aarushi"	ASCII/Unicode codes
Images	Photos	Matrix of pixel values
Sound	Audio	Digital samples
Video	MP4	Sequence of images + audio

Human Language vs. Computer Language

- **Analogy: Talking to a Foreigner Who Doesn't Know Your Language**
  - Imagine you want to talk to a person who **only understands signals (0 and 1)**.
    - Humans speak in **words and sentences**
    - Computers "speak" in **binary code**
  - So NLP acts like a **translator** between human language and computer language.

What NLP Does?

- **Analogy: An Interpreter in a Multilingual Meeting**
- In a conference where people speak English, Nepali, Chinese, etc., an interpreter:
  - Listens to what someone says
  - Understands the meaning
  - Converts it into a language others can understand
- NLP does the same:
  - Reads text/speech
  - Interprets structure & meaning
  - Produces output the computer can use

How NLP Processes Language (Step-by-Step)

- **Analogy: Processing a Sentence Like a Teacher Grades Essays**
- A teacher:
  - **Breaks** the essay into sentences → *(Tokenization)*
  - **Checks** grammar → *(Syntax analysis)*
  - **Understands** meaning → *(Semantics)*
  - **Finds** keywords/important ideas → *(Information extraction)*
  - **Evaluates** or responds → *(Generation/Classification)*
- NLP follows the same pipeline.
  - Tokenization
  - POS tagging
  - Parsing
  - Semantics
  - Named Entity Recognition
  - Sentiment Analysis
  - Text generation
  - Machine Translation
  - Speech Recognition and synthesis

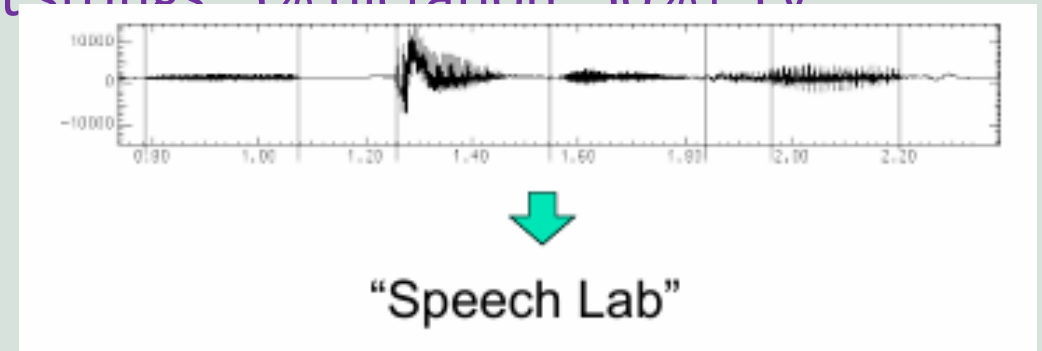
Text Data is Superficial

- An iceberg is a large piece of freshwater ice that has broken off from a snow-formed glacier or ice shelf and is floating in open water.

- … But Language is Complex

Speech Systems

- Automatic Speech Recognition (ASR)
  - Audio in, text out
  - SOTA: 0.3% error for digit strings, 5% dictation, 50%+ TV



- Text to Speech (TTS)
  - Text in, audio out
  - SOTA: totally intelligible (if somemes unnatural)

Example: Siri

Image: Wikipedia

- Siri contains
  - Speech recognition
  - Language analysis
  - Dialog processing
  - Text to speech

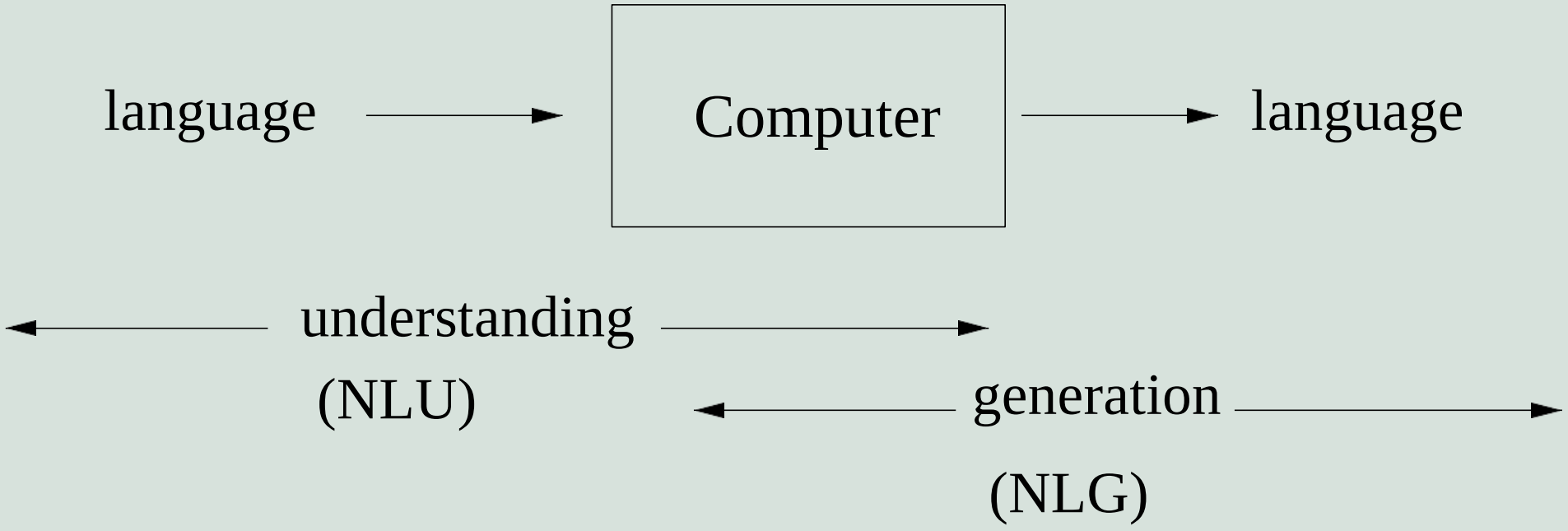
NLP

- **Natural Language Processing (NLP)** is a technology that helps computers *understand, interpret, and use human language* like English or Nepali.
- It is how computers learn to:
  - read text
  - listen to speech
  - understand meaning
  - and respond like humans
- **NLP is the way computers understand and work with human language.**

Natural Language Processing

- is a **field of computer science and artificial intelligence focused on enabling computers to understand, interpret, and generate human language in a way** that is meaningful.
- The goal of NLP is to **bridge the gap** between human communication and computer understanding, allowing computers to process, analyze, and even respond to text or speech in ways that are valuable and useful.

- computers using natural language as input and/or output



Origin of NLP

- NLP originated in the 1950s with rule-based systems and has evolved through statistical methods, machine learning, and deep learning into today's powerful transformer-based models like GPT.
- NLP began when scientists first tried to make computers **understand human language**.
- Its development can be divided into a few important stages:

1. 1950s – The Beginning

- This is when the idea of NLP started.
- **Key Events:**
  - **Alan Turing (1950)**: Proposed the *Turing Test* — a test to check if machines can imitate human conversation.
  - Early goals:
    - Make computers translate languages automatically (like English ↔ Russian).
    - Understand simple sentences.
  - This period focused on **rule-based systems** (hand-written grammar rules).

2. 1960s – Early Language Programs

- **ELIZA (1966)**: First chatbot created by Joseph Weizenbaum.
  It mimicked a psychologist by using simple pattern matching.
- **SHRDLU (late 1960s)**: A program that understood commands in a small "blocks world".
- These showed that computers *could* handle limited language tasks.

3. 1980s – Statistical NLP

- The shift from rules to **probability and statistics**.
- **What changed?**
  - Instead of writing thousands of rules, computers learned from **data**.
  - Language models used:
    - n-grams
    - Hidden Markov Models (HMMs)
    - Decision trees
  - This made NLP more flexible and scalable.

4. 2000s – Machine Learning Era

- More data + better algorithms.
  - Support Vector Machines (SVMs)
  - Maximum Entropy models
  - Conditional Random Fields (CRFs)
- NLP performance improved in translation, tagging, parsing, etc.

5. 2010s – Deep Learning Revolution

- This is when NLP became truly powerful.
- **Major breakthroughs:**
  - **Word embeddings** (Word2Vec, GloVe)
  - **Recurrent Neural Networks (RNNs)**
  - **LSTMs & GRUs**
  - **Sequence-to-sequence models**
  - **Neural Machine Translation (NMT)**
- This enabled systems like Google Translate to dramatically improve.

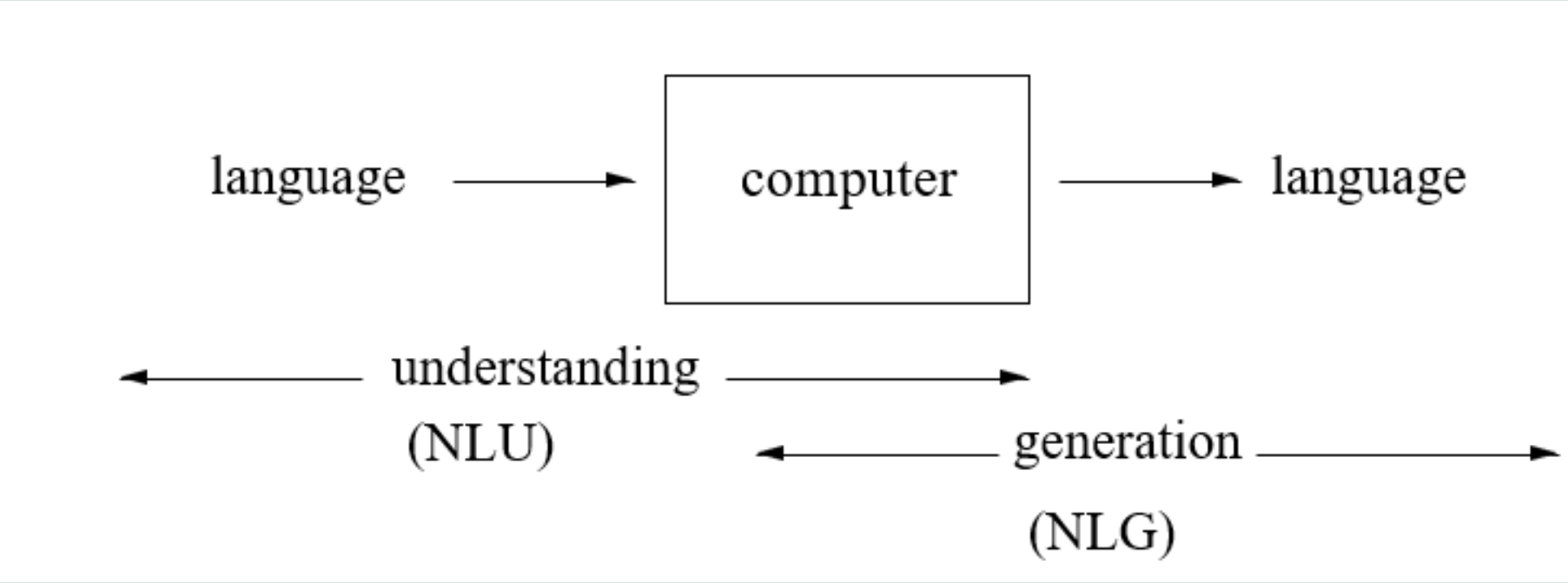
6. 2018–Present – Transformer Era (Modern NLP)

- The biggest change ever in NLP.
- **Key inventions:**
  - **Transformer model (2017)**
  - **BERT (2018)**
  - **GPT models (2018–present)**
- Transformers understand language in a deeply contextual way, leading to:
  - ChatGPT
  - DeepL
  - Google Bard
  - Multilingual understanding
  - High-quality summarization, translation, reasoning
- This era marks the rise of **Generative AI**.

Components of NLP

a) Natural Language Understanding (NLU)

b) Natural Language Generation (NLG)



a) Natural Language Understanding (NLU)

- focuses on the machine's ability to understand and interpret human language.
- NLU involves processes that help in identifying context, semantics, and syntactical structures in text, transforming unstructured data into a structured format that machines can analyze.
- Tasks performed:
  - Tokenization
  - Morphological/Syntactic analysis
  - POS tagging
  - Named Entity Recognition
  - Semantic analysis

- Applications of NLU:
  - **Chatbots and Virtual Assistants**:
    - To interpret user input and respond appropriately.
  - **Sentiment Analysis**:
    - For social media monitoring, customer feedback, and brand perception analysis.
  - **Document Classification**:
    - Categorizing documents by topics, such as news articles, legal documents, or support tickets.
  - **Information Extraction**:
    - Extracting specific information from large text datasets, such as summarizing news articles.

b) Natural Language Generation (NLG)

- the process of creating natural language text from structured or semi-structured data.
- crucial for applications where machines need to produce coherent, contextually relevant, and human-like text, whether as summaries, reports, answers, or dialogue responses.
- Tasks performed:
  - **Content determination:**
    - Decides what information needs to be included in the output text based on the data input and task requirements.
  - **Document Planning**
    - Organizes the overall structure of the generated text, deciding the order and layout of content to improve readability and coherence.
  - **Microplanning**
    - Selects specific words, phrases, and sentence structures to accurately represent the intended content.
  - **Surface Realization**
    - Converts the planned text into a final, grammatically correct output, taking into account syntax and morphology.
  - **Aggregation**
    - Combines similar pieces of information to avoid redundancy and make the text more concise.
  - **Referring Expression Generation**

- **Applications of NLG**
  - **Automated Report Generation**:
    - Used in finance, healthcare, and journalism to produce summaries and detailed reports based on data.
  - **Customer Support Responses**:
    - Automatically generating responses for frequently asked questions or customer inquiries.
  - **Summarization**:
    - Summarizing long documents or articles to extract key points.
  - **Dialogue Systems**:
    - Used in conversational agents to generate human-like responses based on user queries.
  - **Product Descriptions**:
    - Generating descriptions based on product attributes for e-commerce websites.

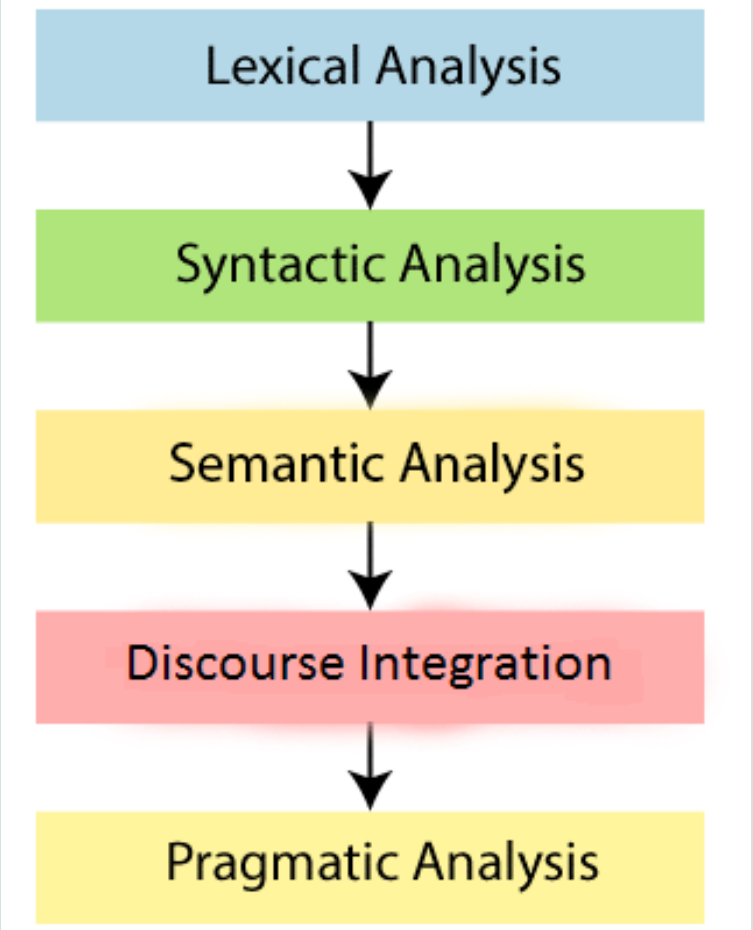
Significance of NLP

1. **Enables Human–Computer Interaction**
   - NLP allows computers to understand and respond to human language.
   - Examples: Siri, Alexa, Google Assistant, chatbots.
2. **Automates Text-Based Tasks**
   - Large volumes of text (emails, documents, reviews) can be processed automatically.
   - Saves time and reduces human effort in customer support, summarization, and documentation.
3. **Extracts Useful Insights from Unstructured Data**
   - Around 80% of data today is unstructured (text, social media, emails).
   - NLP helps identify patterns, sentiments, keywords, and trends from such data.
4. **Improves Search and Information Retrieval**
   - Search engines use NLP to understand queries, rank results, and provide relevant answers.
   - Autocomplete, spell correction, and question-answering systems rely heavily on NLP.
5. **Enhances Communication Across Languages**
   - Machine translation (Google Translate, DeepL) breaks language barriers.

6. **Supports Decision-Making**
   - NLP systems analyze reports, customer feedback, surveys, and news to support business or policy decisions.
   - Used in finance, health, governance, and research.
7. **Personalization and Recommendation**
   - NLP helps systems understand user preferences from text or voice inputs.
   - Used in personalized learning platforms, e-commerce, and entertainment apps.
8. **Helps in Sentiment and Emotion Analysis**
   - Identifies opinions (positive/negative/neutral) in tweets, reviews, comments.
   - Crucial for marketing, political analysis, and customer experience management.
9. **Makes Education More Accessible**
   - Speech-to-text and text-to-speech improve accessibility for visually or hearing-impaired learners.
   - Automated grading, grammar correction, and personalized tutoring are enhanced by NLP.
10. **Advances Artificial Intelligence**
    - NLP is core to AGI-like systems, digital assistants, chatbots, and intelligent agents.
    - It helps machines reason, converse, and learn from human text data.

Phases in NLP

- There are phases in NLP which need to be performed in order to extract meaningful information from the text corpus.
- Once these phases are completed, you are ready with your refined text and then you can apply some machine learning model to predict something



1. Lexical Analysis:

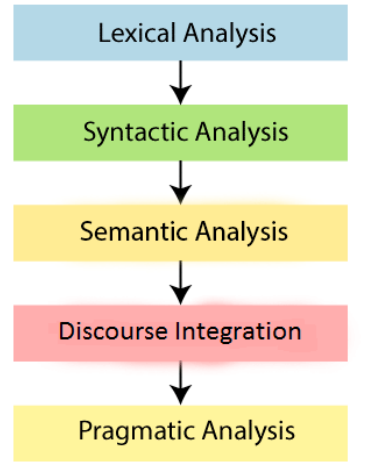
- In this phase, the **text is broken down into paragraphs, sentences and words**.
- Analysis is done for identification and description of the structure of words.
- It includes techniques as follows:
  - **Stop word removal** (removing ‘and’, ‘of’, ‘the’ etc. from text)
  - **Tokenization** (breaking the text into sentences or words)
    - Word tokenizer
    - Sentence tokenizer
    - Tweet tokenizer
  - **Stemming** (removing ‘ing’, ‘es’, ‘s’ from the tail of the words)
  - **Lemmatization** (converting the words to their base forms)

Input sentence:
"NLP is fascinating!“

Lexical analysis (tokenization) might produce:
["NLP", "is", "fascinating", "!"]

2. Syntactic Analysis (Parsing)

- In this phase, the sentence is **checked whether it is well-formed or not.**
- It involves **analyzing the grammatical structure of a sentence** to determine how words are related to each other.
- Syntactic Analysis is used to check grammar, word arrangements, and shows the relationship among the words.
- For example:
  - **Correct Syntax:** *Sun rises in the east.*
  - **Incorrect Syntax:** *Rise in sun the east.*
- Some of the techniques used in this phase are:
  - Dependency Parsing
  - Parts of Speech (POS) tagging



3. Semantic Analysis (Meaning Analysis)

- This phase gives meaning to the sentence.
- It understands what each word and sentence means in context.
- **Example:**
- **Sentence**:
  "John gave Mary a book."
- **Semantic analysis** understands:
  - *John* is the **giver**
  - *Mary* is the **receiver**
  - *Book* is the **item being transferred**
- Example:– “Ram went to the bank.”– Does “bank” mean river bank or money bank?

4. Discourse Integration

- It connects the current sentence with the previous sentences to make sense of the whole conversation or paragraph.
- In this phase, the impact of the sentences before a particular sentence and the effect of the current sentence on the upcoming sentences is determined.
- Example:– “Aarushi bought a laptop. She loves it.”
- Here, “she” refers to Aarushi.

5. Pragmatic Analysis

- Sometimes the discourse integration phase and pragmatic analysis phase are combined.
- There can be multiple scenarios where the intent of a sentence can be misunderstood if the machine doesn't have real world knowledge.
- This phase interprets the intended meaning of a sentence in a real-world context — tone, situation, and purpose.
- Example:
  - "Can you open the window?"
  - Not a question about ability
  - It's a polite request.

Why NLP is difficult?

- “At last, a computer that understands you like your mother”

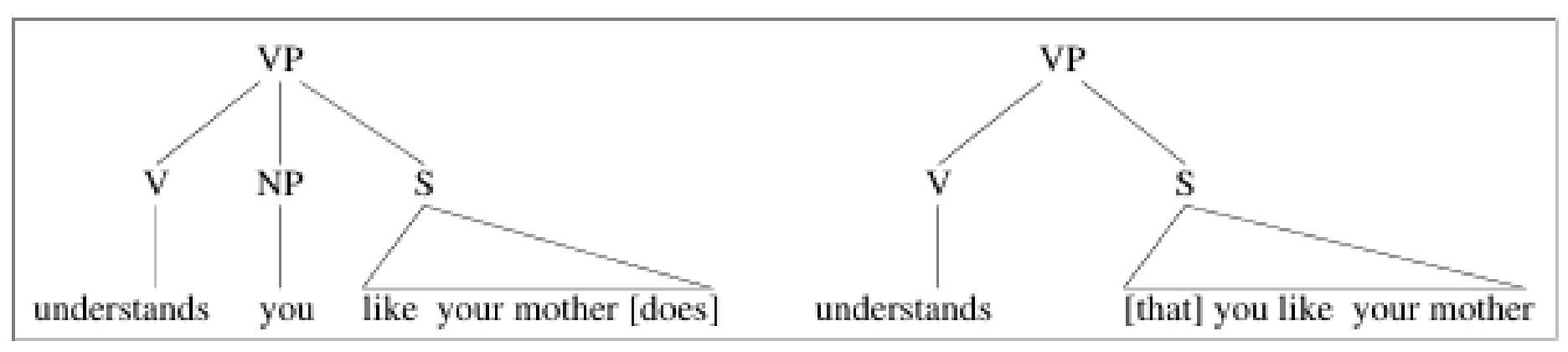
  1. It understands you as well as your mother understands you
  2. It understands (that) you like your mother
  3. It understands you as well as it understands your mother.

- 1 and 3: Does this mean well, or poorly?

Ambiguity at Many Levels

- At the acoustic level (speech recognition):
  1. “ . . . a computer that understands you like your mother”
  2. “ . . . a computer that understands you lie cured mother”

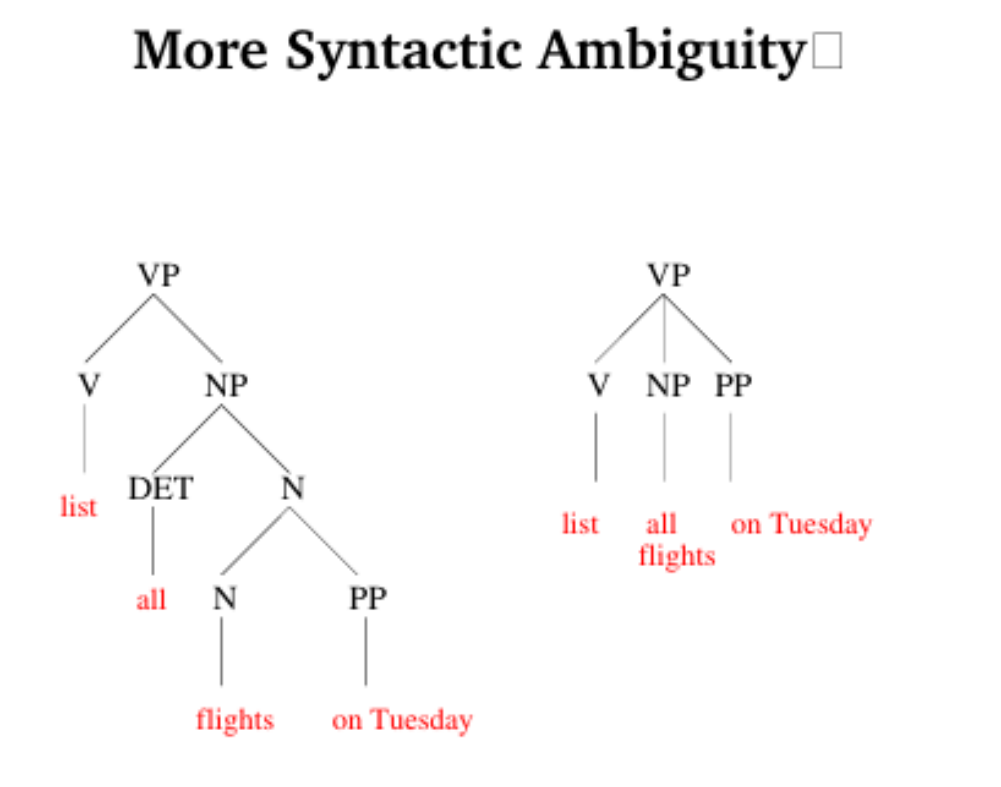
- **At the <u>syntactic level:</u>**



- Different structures lead to different interpretations.
- The ambiguity arises because the sentence can be parsed in two ways:
- **As a verb with two complements:**
  - NP ("you") and S ("like your mother").
  - Gives a *comparative/ manner* meaning.
- **As a verb with a single clausal complement:**
  - S = "you like your mother".
  - Gives a *that-clause* (content clause) meaning.

Symbol	Meaning	Example in the diagram
NP	Noun Phrase	*you*
VP	Verb Phrase	*understands you like your mother*
S	Clause (Sentence)	*you like your mother*

- The image shows another example of **syntactic ambiguity**, using the sentence:
- “List all flights on Tuesday.”
  - **List all the flights *that occur on Tuesday*.**
    - Flights occur on Tuesday
  - On Tuesday, list all flights.
    - The listing should be done on Tuesday



- **At the <u>semantic (meaning) level</u>:**
- Ambiguity does not occur only in syntax. It appears at **many linguistic levels**, including **meaning (semantic)** levels.
- A single word can have more than one meaning.
  This is called **word sense ambiguity** or **lexical ambiguity**.
- **Example: the word "mother".**
- It has **two very different meanings**:
  - a woman who has given birth to a child
  - a stringy slimy substance consisting of yeast cells and bacteria; is added to cider or wine to produce vinegar
- This is an instance of word sense ambiguity. These two meanings have **no relationship**, but the same word is used for both.

- “They put money in the bank.”
  - They deposited money.
  - They buried money in the mud on a riverbank.
- This is **lexical ambiguity** because the noun **bank** has more than one sense.

- **<u>Discourse-Level Ambiguity</u>:**
- At the **discourse level** (multi-clause or multi-sentence context), ambiguity often arises through **anaphora**.
- Anaphora occurs when a word (usually a pronoun like *he, she, it, they*) refers back to some entity previously mentioned in the discourse.
- Sometimes it is **unclear which earlier entity the pronoun refers to**, creating ambiguity.
- **Example:** Alice says they've built a computer that understands you like your mother. But she…
- What does **she** refer to?
- It can refer to **two different discourse entities**:
  - Interpretation 1 — she = Alice
  - Interpretation 2 — she = your mother
- Both are possible based on the previous sentence, making the pronoun **ambiguous** at the discourse level.

Applications of NLP

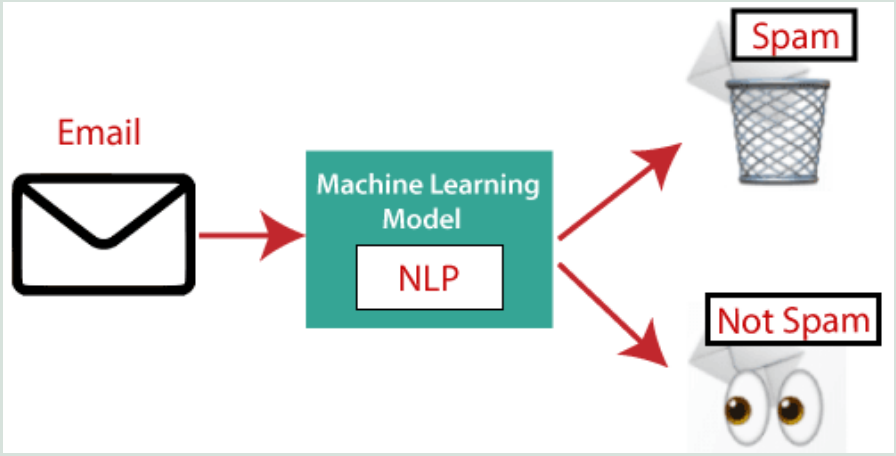
1. Question Answering
2. Spam detection
3. Sentiment Analysis
4. Machine translation
5. Spelling correction
6. Speech recognition
7. Chatbot
8. Information Retrieval
9. And many more

1. Question Answering

- Question Answering focuses on building systems that automatically answer the questions asked by humans in a natural language.

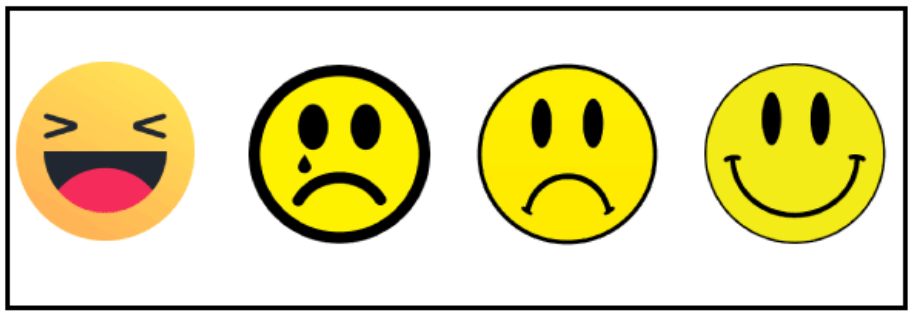
2. Spam detection

- is used to detect unwanted e-mails getting to a user's inbox.



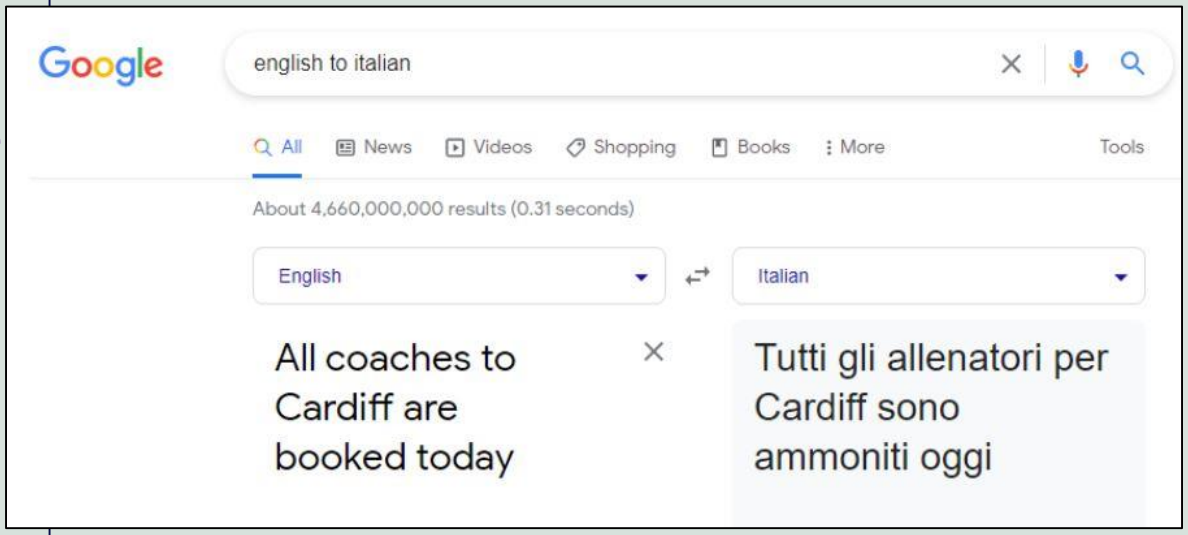
3. Sentiment Analysis

- Sentiment Analysis is also known as **opinion mining**.
- It is used on the web to analyze the attitude, behavior, and emotional state of the sender.
- This application is implemented through a combination of NLP and statistics by assigning the values to the text (positive, negative, or natural), identify the mood of the context (happy, sad, angry, etc.)



4. Machine Translation

- Machine translation is used to translate text or speech from one natural language to another natural language.
- **Example:** Google Translator



5. Spelling correction

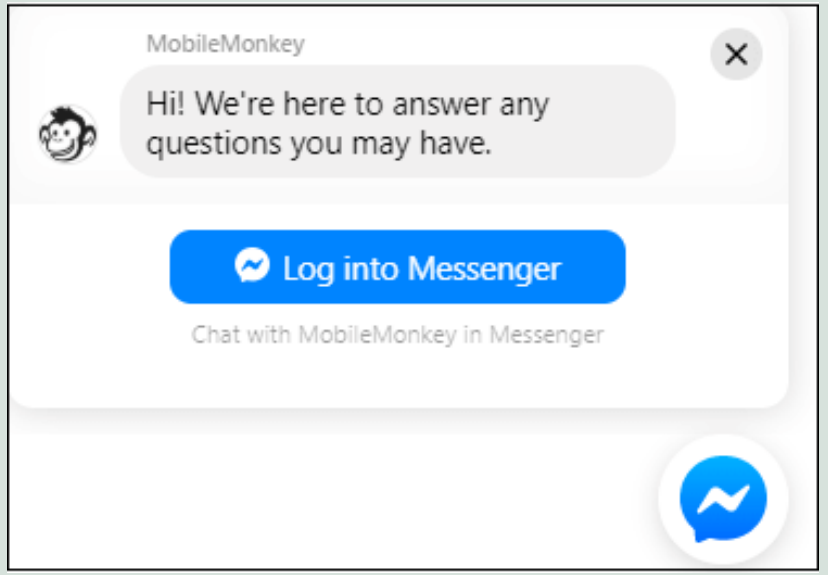
- Microsoft Corporation provides word processor software like MS-word, PowerPoint for the spelling correction

6. Speech Recognition

- Speech recognition is used for converting spoken words into text.
- It is used in applications, such as mobile, home automation, video recovery, dictating to Microsoft Word, voice biometrics, voice user interface, and so on.

7. Chatbot

- Implementing the Chatbot is one of the important applications of NLP.
- It is used by many companies to provide the customer's chat services.



8. Information Retrieval

- Information retrieval involves returning a set of documents in response to a user query: Internet search engines are a form of IR.
- Information extraction involves trying to discover specific information from a set of documents.
- However, one change from classical IR is that Internet search now uses techniques that rank documents according to how many links there are to them (e.g., Google's PageRank) as well as the presence of search terms.
- NLP (Natural Language Processing) plays a crucial role in information retrieval by enhancing the search and retrieval process of relevant information from large text collections.

Assignment:

- Challenges in NLP

Popular NLP tools and libraries:

1. NLTK (Natural Language Toolkit)
2. SpaCy
3. Hugging Face Transformers
4. Gensim
5. Stanford NLP (Stanza)
6. TextBlob
7. AllenNLP

1. NLTK (Natural Language Toolkit)

- **Overview**
  - **Developed by**: Steven Bird and Edward Loper at the University of Pennsylvania.
  - **Languages**: Primarily Python.
  - **Audience**: Ideal for beginners and educational purposes.
- **Key Features**
  - Comprehensive collection of corpora, lexical resources, and pre-built NLP tasks.
  - Wide range of functionalities including tokenization, POS tagging, stemming, lemmatization, parsing, and semantic reasoning.
  - Extensive documentation and beginner-friendly, making it great for academic and learning purposes.
- **Limitations**
  - Slower performance for large datasets.
  - Less support for deep learning or modern NLP techniques.
- **Example Use Case**
  - Analyzing the sentiment of text or performing basic linguistic analysis on news articles, social media posts, or academic text.

2. SpaCy

- **Overview**
  - **Developed by**: Explosion AI.
  - **Languages**: Python.
  - **Audience**: Geared towards production-ready and real-world applications.
- **Key Features**
  - Highly efficient and optimized for performance, especially with large datasets.
  - Supports deep learning through integration with libraries like PyTorch and TensorFlow.
  - Built-in support for entity recognition, dependency parsing, and part-of-speech tagging.
  - Provides pre-trained models in multiple languages.
  - Allows users to add custom models or integrate with transformer-based architectures (e.g., BERT, RoBERTa).
- **Limitations**
  - Limited flexibility for customizations compared to more open libraries.
  - Focuses on efficiency and speed, which might limit academic exploration of certain NLP areas.
- **Example Use Case**
  - Named Entity Recognition (NER) in documents, real-time text processing in chatbots, and sentiment analysis in large datasets.

3. Hugging Face Transformers

- **Overview**
  - **Developed by**: Hugging Face.
  - **Languages**: Python.
  - **Audience**: Suited for researchers and developers working on cutting-edge NLP.
- **Key Features**
  - Supports state-of-the-art transformer models like BERT, GPT, T5, RoBERTa, and many others.
  - Integrates with PyTorch and TensorFlow, allowing for seamless deep learning applications.
  - Contains over 10,000 pre-trained models that can be fine-tuned for various NLP tasks.
  - Supports tasks such as text classification, summarization, translation, question answering, and more.
  - Extensive documentation, tutorials, and a large community.
- **Limitations**
  - Requires significant computational resources, especially for large transformer models.
  - Beginners may find it challenging to configure models without some experience with deep learning frameworks.
- **Example Use Case**
  - Developing an advanced question-answering system, fine-tuning a BERT model for a specific classification task, or creating a language generation tool.

4. Gensim

- **Overview**
  - **Developed by**: Radim Řehůřek.
  - **Languages**: Python.
  - **Audience**: Primarily used for topic modeling and document similarity tasks.
- **Key Features**
  - Specialized in document similarity analysis, topic modeling, and vector-space modeling.
  - Efficient for handling large corpora and unsupervised text analysis.
  - Implements popular models like Word2Vec, FastText, and Doc2Vec for creating word embeddings.
  - Good for creating semantic representations and finding hidden topics in text data.
- **Limitations**
  - Less suited for tasks like sentiment analysis or named entity recognition.
  - Primarily focuses on unsupervised and semi-supervised tasks rather than fully supervised learning.
- **Example Use Case**
  - Clustering news articles into different topics, analyzing themes in a collection of customer reviews, or finding similarity between documents.

5. Stanford NLP (Stanza)

- **Overview**
  - **Developed by**: Stanford University.
  - **Languages**: Python, Java.
  - **Audience**: Suited for linguistically rich NLP tasks and multi-language support.
- **Key Features**
  - Provides pre-trained models for over 70 languages.
  - Strong capabilities in syntactic analysis, dependency parsing, and named entity recognition.
  - Initially developed in Java (CoreNLP), with Stanza being the newer Python library for easy integration with Python environments.
  - Incorporates linguistic resources to ensure high accuracy in parsing and named entity recognition.
- **Limitations**
  - Slower performance compared to SpaCy or Hugging Face Transformers.
  - Requires additional computational resources for specific tasks due to deep linguistic analysis.
- **Example Use Case**
  - Parsing complex sentences for syntactic structure, multi-language text processing, and extracting detailed linguistic information from text.

6. TextBlob

- **Overview**
  - **Developed by**: Steven Loria.
  - **Languages**: Python.
  - **Audience**: Suitable for beginners and for rapid prototyping.
- **Key Features**
  - Built on top of NLTK and Pattern, offering a simplified interface for NLP tasks.
  - Easy-to-use API for text classification, part-of-speech tagging, noun phrase extraction, and sentiment analysis.
  - Simplifies common NLP tasks for quick prototyping and experimentation.
- **Limitations**
  - Limited functionality for advanced NLP applications.
  - Slower and less efficient for large-scale production use.
- **Example Use Case**
  - Building simple sentiment analysis tools or small-scale text classification applications.

7. AllenNLP

- **Overview**
  - **Developed by**: Allen Institute for AI.
  - **Languages**: Python.
  - **Audience**: Aimed at researchers and practitioners working on innovative NLP research.
- **Key Features**
  - Built with deep learning frameworks, especially PyTorch, making it flexible for custom model building.
  - Includes models for reading comprehension, textual entailment, coreference resolution, and more.
  - Provides modular components and a configuration-driven approach for custom NLP research tasks.
  - Actively maintained with strong community support and contributions from NLP researchers.
- **Limitations**
  - High learning curve for beginners.
  - Requires significant computational resources.
- **Example Use Case**
  - Developing and training custom neural network models for research purposes, such as a model for understanding textual entailment or coreference resolution in documents.

End of chapter