**5. Artificial Intelligence (AI)**

**Understanding AI**

Artificial Intelligence (AI) is a branch of computer science that enables machines to mimic human intelligence. AI applications range from simple rule-based systems to complex deep learning models used in natural language processing, robotics, and decision-making.

**Real-time Example:**

Virtual assistants like Siri and Alexa use AI to understand and respond to user queries by processing natural language.

**Module 1: Introduction to Text Mining and NLP**

**Overview of Text Mining**

Text Mining is the process of extracting meaningful information from unstructured text data. It involves techniques such as information retrieval, pattern recognition, and machine learning to analyze text-based data.

**Real-time Example:**

A customer service chatbot analyzes emails to detect common issues such as "billing problems" or "technical support." By applying text mining, companies can categorize and prioritize customer queries automatically.

**Need for Text Mining**

In today’s world, data is primarily available in text format (emails, blogs, social media posts, medical records). Traditional data analysis techniques fail to handle unstructured text data efficiently, making text mining essential for businesses and research.

**Real-time Example:**

A marketing team wants to understand customer sentiment from online product reviews. Using text mining, they can classify reviews as positive, negative, or neutral to improve their products.

**Natural Language Processing (NLP) in Text Mining**

NLP allows computers to understand, interpret, and manipulate human language. It includes tasks such as tokenization, stemming, lemmatization, and named entity recognition (NER).

**Real-time Example:**

A news aggregator platform uses NLP to summarize long news articles into short snippets while maintaining the main ideas, making content easier to read.

**Applications of Text Mining**

* **Sentiment Analysis:** Understanding public opinion on social media.
* **Spam Detection:** Filtering out unwanted spam emails.
* **Chatbots and Virtual Assistants:** AI-driven assistants like Siri and Alexa process natural language queries.
* **Healthcare Text Analysis:** Extracting critical information from medical records for diagnosis.
* **Financial Fraud Detection:** Detecting fraudulent transactions using transaction descriptions.

**Real-time Example:**

A bank uses text mining to detect fraudulent activities by analyzing transaction descriptions. If a pattern of suspicious activity is found, the bank flags it for investigation.

**OS Module in Python**

The OS module provides functions for interacting with the operating system, such as reading and writing files, accessing environment variables, and managing directories.

**Real-time Example:**

A data analyst wants to automate file management. Using the OS module, they can create a script to organize files into different folders based on their formats.

**Reading, Writing to Text and Word Files**

Python allows users to read and write .txt and .docx files, making it useful for handling document-based data.

**Real-time Example:**

A research team is processing survey responses stored in .docx files. Using Python, they extract responses and analyze the data for trends.

**Setting the NLTK Environment**

NLTK (Natural Language Toolkit) is a powerful Python library for NLP. It provides tools for text processing, classification, tokenization, and stemming.

**Real-time Example:**

An HR department uses NLTK to analyze job descriptions and match them with suitable candidates based on keyword extraction.

**Accessing the NLTK Corpora**

NLTK includes various text corpora (datasets) such as:

* stopwords (common words like "the", "is", "in")
* wordnet (dictionary for synonyms and antonyms)
* gutenberg (collection of classic books)

**Real-time Example:**

A linguistics researcher uses NLTK’s wordnet to find synonyms and improve a translation tool’s accuracy.

**Hands-on:**

1. **Install NLTK Packages using NLTK Downloader**
   * Run import nltk and nltk.download('all') to install required datasets.
   * *Example:* Downloading stopwords to remove unnecessary words from a dataset.
2. **Accessing Your Operating System using the OS Module**
   * Using os.listdir() to list all files in a directory.
   * *Example:* Automating file organization by moving log files into a separate folder.
3. **Reading JSON and PKL Files**
   * JSON files store data as key-value pairs.
   * PKL (Pickle) files serialize Python objects.
   * *Example:* Loading and saving chatbot training data using .json and .pkl files.

**Skills You Will Learn:**

* Reading & Writing .txt Files from/to Local Storage
* Reading & Writing .docx Files from/to Local Storage
* Working with NLTK Corpora to analyze text data
* Automating tasks using Python’s OS module
* Handling structured (JSON) and unstructured (text) data

**How This Guide Helps in Interviews:**

* Provides a clear understanding of text mining and NLP.
* Demonstrates real-world applications useful in various industries.
* Prepares you to answer practical interview questions related to AI-driven text processing.
* Equips you with hands-on skills in Python for text analysis.

**Module 2: Text Analytics**

Text Analytics is a crucial aspect of Natural Language Processing (NLP) used to extract meaningful insights from textual data. It plays a vital role in sentiment analysis, chatbot development, and search engine optimization.

**1. Tokenization**

**Definition:** Tokenization is the process of breaking down a text into smaller units (tokens), such as words, phrases, or sentences.

**Real-Time Example:**

* In a chatbot system, tokenization helps break user input into words for better understanding, e.g., converting "How is the weather today?" into ["How", "is", "the", "weather", "today", "?"]

**2. Frequency Distribution**

**Definition:** It calculates the frequency of each word appearing in a given text, helping identify common words.

**Real-Time Example:**

* In marketing, analyzing product reviews to determine frequently used words like "good," "bad," or "expensive" helps understand customer sentiment.

**3. Different Types of Tokenizers**

**Definition:** Tokenizers vary based on how they split text—word tokenizers, sentence tokenizers, and regex-based tokenizers.

**Real-Time Example:**

* A search engine uses different tokenizers to improve search results. A regex tokenizer can remove unwanted characters, making searches more accurate.

**4. Bigrams, Trigrams & Ngrams**

**Definition:** These are continuous sequences of words:

* **Bigram:** Two consecutive words (e.g., "machine learning")
* **Trigram:** Three consecutive words (e.g., "artificial intelligence system")
* **Ngrams:** General form for n consecutive words

**Real-Time Example:**

* In predictive text applications (e.g., smartphone keyboards), n-grams help suggest the next word while typing based on common sequences.

**5. Stemming**

**Definition:** Stemming reduces words to their root form by removing suffixes.

**Real-Time Example:**

* A search engine indexing system will treat "running," "runs," and "ran" as "run" to improve search efficiency.

**6. Lemmatization**

**Definition:** Lemmatization converts words to their base or dictionary form while maintaining meaning.

**Real-Time Example:**

* In AI-driven news categorization, lemmatization helps classify words like "better" as "good" for better topic grouping.

**7. Stopwords**

**Definition:** Stopwords are common words (e.g., "the", "is", "in") that are removed to improve processing efficiency.

**Real-Time Example:**

* In sentiment analysis, removing stopwords helps focus on meaningful words like "happy," "amazing," or "bad."

**8. POS Tagging (Part of Speech Tagging)**

**Definition:** Identifies each word's grammatical category (noun, verb, adjective, etc.).

**Real-Time Example:**

* AI-powered grammar checkers use POS tagging to suggest correct sentence structures.

**9. Named Entity Recognition (NER)**

**Definition:** Identifies and categorizes entities such as names, locations, and dates.

**Real-Time Example:**

* In financial applications, NER helps extract company names, stock symbols, and locations from news articles to automate market analysis.

**Hands-on Applications**

* **Regex, Word, Blankline, Sentence Tokenizers:** Extracting structured data from text.
* **Bigrams, Trigrams & Ngrams:** Predictive text input for messaging apps.
* **Stopword Removal:** Enhancing search engine relevance.
* **UTF Encoding & Handling URLs/Hashtags:** Cleaning text data from social media posts.
* **POS Tagging & NER:** Automated chatbots understanding user intents.

**Skills You Will Learn:**

* Tokenization
* Stopword Removal
* UTF Encoding
* POS Tagging
* Named Entity Recognition (NER)

**How This Guide Helps in Interviews:**

* Provides clear explanations with real-world examples.
* Helps answer conceptual and applied interview questions.
* Prepares you for hands-on AI and NLP-based tasks.

**Module 3: Analyzing Sentence Structure**

**Understanding Sentence Structure in AI**

Natural Language Processing (NLP) enables machines to understand and process human language. One critical aspect of NLP is sentence structure analysis, which helps in various applications like machine translation, chatbots, and text summarization.

**Key Topics and Their Explanations**

**1. Syntax Trees**

A syntax tree is a hierarchical representation of the grammatical structure of a sentence. It breaks down sentences into their individual components such as nouns, verbs, adjectives, and phrases.

**Real-time Example:** A grammar-checking tool like Grammarly analyzes your sentence structure using syntax trees to suggest corrections.

**2. Chunking**

Chunking is a process in NLP that groups words together into meaningful phrases, such as noun phrases (NP) and verb phrases (VP).

**Real-time Example:** In a resume parser, chunking helps extract job titles, company names, and key responsibilities by identifying noun and verb phrases.

**3. Chinking**

Chinking is the opposite of chunking; it removes certain patterns from a chunked phrase to refine results.

**Real-time Example:** A chatbot removes unnecessary filler words (like "uh", "hmm") from user queries to improve search accuracy.

**4. Context-Free Grammars (CFG)**

CFG is a set of production rules that define the syntax of a language. It is widely used in NLP for parsing and language modeling.

**Real-time Example:** Virtual assistants like Siri and Alexa use CFGs to parse and understand spoken commands efficiently.

**5. Automating Text Paraphrasing**

Text paraphrasing involves rewording sentences while preserving meaning. AI-driven paraphrasing tools use deep learning models along with syntax trees and CFGs to generate alternative versions of text.

**Real-time Example:** Tools like QuillBot use AI to rewrite and paraphrase content while maintaining coherence and grammar.

**Hands-on Practice for Interviews**

* **Parsing Syntax Trees:** Breaking down sentences into their syntactic components using NLP libraries like nltk.
* **Chunking & Chinking:** Extracting useful phrases from text using regular expressions.
* **Automating Text Paraphrasing using CFGs:** Creating rule-based text transformation models for language applications.

**Skills You Will Learn**

✔ Chunking for phrase identification  
✔ Chinking for text refinement  
✔ Automating text paraphrasing for NLP applications

**Module 4: Text Classification-I**

**Introduction to Text Classification**

Text classification is a Natural Language Processing (NLP) technique used to categorize text into predefined labels. It is widely used in spam detection, sentiment analysis, and document categorization.

**Machine Learning: Brush Up**

Before diving into text classification, it's essential to have a basic understanding of machine learning concepts such as supervised learning, classification algorithms, and model evaluation metrics.

**Real-time Example:** A company wants to filter spam emails from genuine ones. Using machine learning, they train a model to classify emails as either spam or non-spam.

**Bag of Words (BoW) Approach**

BoW is a simple representation of text data where each word is treated as a separate feature. The frequency of words is counted without considering grammar or word order.

**Real-time Example:** An online review system uses BoW to analyze customer feedback. Common words like "excellent," "poor," and "average" help determine product ratings.

**Count Vectorizer**

Count Vectorizer is a method that converts text into a matrix of token counts, helping machines understand textual data in numerical form.

**Real-time Example:** A movie review website uses CountVectorizer to analyze the frequency of positive and negative words in user reviews to determine sentiment.

**Term Frequency (TF)**

TF measures how frequently a word appears in a document. It helps in identifying important terms within a text.

**Real-time Example:** A news website highlights trending topics by identifying frequently occurring words in recent articles using TF.

**Inverse Document Frequency (IDF)**

IDF is used to give higher importance to rare words and lower importance to common words across multiple documents.

**Real-time Example:** A search engine like Google ranks search results by giving priority to unique keywords that are less frequent but more meaningful using IDF.

**Hands-on Demonstration**

1. **Demonstrate Bag of Words Approach** – Applying BoW to convert text data into numerical form.
2. **Working with CountVectorizer()** – Implementing CountVectorizer to transform text into feature vectors.
3. **Using TF & IDF** – Applying TF-IDF to enhance text representation by weighting terms based on their relevance.

**Skills You Will Learn**

* Understanding the **Bag of Words** model for text representation.
* Using **CountVectorizer()** to convert text into numerical data.
* Applying **TF-IDF** to improve text classification performance.

**Module 5: Introduction to Deep Learning**

**What is Deep Learning?**

Deep Learning is a subset of Machine Learning that mimics the way the human brain processes data. It uses artificial neural networks with multiple layers to learn from large amounts of data. Deep Learning is widely used in image recognition, speech processing, and autonomous systems.

**Real-time Example:**

Voice assistants like Siri, Google Assistant, and Alexa use Deep Learning models to understand human speech and respond accurately.

**Curse of Dimensionality**

As the number of features (dimensions) in a dataset increases, the computational complexity also grows, making it difficult to process data efficiently. High-dimensional data can lead to overfitting and increased training time.

**Real-time Example:**

In an e-commerce website, predicting customer preferences based on thousands of parameters (age, location, past purchases, clicks) can be inefficient and lead to irrelevant recommendations. Feature selection or dimensionality reduction techniques like PCA help in such cases.

**Machine Learning vs. Deep Learning**

* **Machine Learning:** Uses structured data and requires feature engineering.
* **Deep Learning:** Works with unstructured data and learns features automatically using multiple layers of neural networks.

**Real-time Example:**

Traditional Machine Learning models require manual selection of key features in medical diagnosis, whereas Deep Learning can analyze complex MRI scans without human intervention, improving accuracy.

**Use Cases of Deep Learning**

* **Healthcare:** Disease prediction using medical images.
* **Finance:** Fraud detection in credit card transactions.
* **Retail:** Personalized product recommendations.
* **Self-Driving Cars:** Object detection and route optimization.
* **Entertainment:** AI-generated movie recommendations.

**Real-time Example:**

Netflix uses Deep Learning to analyze users' watching habits and suggest personalized content.

**Human Brain vs. Neural Network**

* **Human Brain:** Comprises billions of neurons connected to form complex pathways for decision-making.
* **Neural Network:** Uses artificial neurons (perceptrons) to process and classify information.

**Real-time Example:**

Facial recognition in smartphones works by mapping human facial features just like the human brain recognizes faces in daily life.

**What is Perceptron?**

A Perceptron is the simplest type of artificial neural network, consisting of a single layer of neurons. It is used for binary classification tasks.

**Real-time Example:**

Spam email detection uses a perceptron to classify emails as spam or non-spam based on keywords and patterns.

**Learning Rate**

The learning rate determines how quickly the model updates its weights during training. A high learning rate can cause overshooting, while a low rate can slow down learning.

**Real-time Example:**

In training a chatbot, adjusting the learning rate ensures that the model learns optimally from user interactions without making abrupt changes.

**Epoch**

An epoch refers to one complete cycle through the entire dataset during training. More epochs allow better learning but can lead to overfitting.

**Real-time Example:**

Training an image recognition model for self-driving cars requires multiple epochs to ensure high accuracy in detecting traffic signs.

**Batch Size**

Batch size defines the number of training samples processed before updating the model’s weights. Smaller batch sizes generalize better, while larger ones improve training speed.

**Real-time Example:**

In stock market prediction, using an optimal batch size balances quick training and accurate trend analysis.

**Activation Function**

Activation functions introduce non-linearity into neural networks, allowing them to learn complex patterns. Common types include ReLU, Sigmoid, and Tanh.

**Real-time Example:**

ReLU is widely used in image recognition applications like Facebook’s photo tagging feature to detect faces efficiently.

**Single Layer Perceptron**

A single-layer perceptron is the basic building block of neural networks, processing weighted inputs and passing them through an activation function.

**Real-time Example:**

A perceptron can be used in fraud detection to classify transactions as fraudulent or legitimate based on transaction history and user behavior.

**Hands-on: Single Layer Perceptron**

To practically understand a perceptron, one can implement a simple binary classification model for spam email detection using a single-layer neural network.

**Skills You Will Learn**

* Understanding the fundamentals of Deep Learning
* Handling high-dimensional data (Curse of Dimensionality)
* Implementing perceptrons and activation functions
* Adjusting hyperparameters like learning rate, batch size, and epochs
* Real-world applications of Deep Learning in different industries

**How This Guide Helps in Interviews**

* Provides a structured explanation of Deep Learning concepts.
* Includes industry-relevant use cases for better understanding.
* Prepares you with real-world examples to answer interview questions effectively.

**Module 6:  
Getting Started with TensorFlow 2.0**

**Introduction to TensorFlow 2.x**

TensorFlow 2.x is an open-source machine learning framework developed by Google, designed for building and deploying deep learning models. It is widely used for applications in AI, including image recognition, speech processing, and predictive analytics.

**Real-time Example:**

A healthcare company wants to detect pneumonia from X-ray images. Using TensorFlow 2.x, they train a deep learning model to classify X-ray scans as normal or pneumonia-affected.

**Installing TensorFlow 2.x**

TensorFlow can be installed using pip install tensorflow. It supports CPU and GPU acceleration for efficient training.

**Real-time Example:**

A data scientist wants to train a deep learning model on their laptop but later moves it to a cloud GPU environment for faster training.

**Defining Sequence Model Layers**

In TensorFlow 2.x, a neural network consists of sequential layers that process input data.

**Real-time Example:**

An e-commerce site wants to recommend products based on user behavior. A sequential model with embedding layers is used to learn customer preferences.

**Activation Function**

Activation functions introduce non-linearity to neural networks. Common activation functions include ReLU, Sigmoid, and Softmax.

**Real-time Example:**

In sentiment analysis of customer reviews, ReLU is used in hidden layers to extract key features, while Softmax helps classify sentiments as positive, negative, or neutral.

**Layer Types**

Common layer types in TensorFlow 2.x include:

* **Dense Layer:** Fully connected neural network layer.
* **Convolutional Layer:** Used in image recognition.
* **Recurrent Layer:** Used in time-series analysis and NLP.

**Real-time Example:**

A facial recognition app uses Convolutional Layers to detect and classify human faces in photos.

**Model Compilation**

Compiling a model involves specifying the optimizer, loss function, and evaluation metrics.

**Real-time Example:**

A bank wants to predict credit card fraud. They compile a fraud detection model using a binary cross-entropy loss function and an Adam optimizer.

**Model Optimizer**

Optimizers like Adam, SGD, and RMSprop help adjust model weights for better learning.

**Real-time Example:**

A weather prediction system uses the Adam optimizer to improve accuracy in forecasting temperature trends.

**Model Loss Function**

Loss functions measure how well a model’s predictions match actual values. Examples include MSE (for regression) and Categorical Cross-Entropy (for classification).

**Real-time Example:**

An insurance company predicts claim amounts using a neural network trained with the Mean Squared Error loss function.

**Model Training**

Training involves feeding data into the model, adjusting weights, and iterating until accuracy improves.

**Real-time Example:**

A self-driving car is trained on road images to recognize traffic signs using deep learning in TensorFlow.

**Digit Classification using Simple Neural Network in TensorFlow 2.x**

A basic neural network can classify handwritten digits (e.g., MNIST dataset).

**Real-time Example:**

A banking app digitizes handwritten cheques using a trained neural network that recognizes numbers.

**Improving the Model**

**Adding Hidden Layers**

More hidden layers can capture complex patterns in data.

**Real-time Example:**

A medical AI system adds hidden layers to better diagnose rare diseases from MRI scans.

**Adding Dropout**

Dropout prevents overfitting by randomly disabling neurons during training.

**Real-time Example:**

A stock market prediction model uses dropout to avoid making overly confident predictions based on historical trends.

**Using Adam Optimizer**

Adam optimizer adapts learning rates for faster convergence.

**Real-time Example:**

A chatbot’s AI model trains faster with Adam optimizer, improving response times for users.

**Hands-on: Classifying Handwritten Digits using TensorFlow 2.0**

The MNIST dataset is used to train a neural network that recognizes digits.

**Real-time Example:**

An automated form processing system extracts and verifies handwritten zip codes from scanned documents.

**Skills You Will Learn**

* Installing and Working with TensorFlow 2.0
* Defining and Training Neural Networks
* Understanding Activation Functions and Optimizers
* Improving Model Performance with Dropout and Hidden Layers

**Module 7: Convolutional Neural Networks (Optional)**

**Understanding Convolutional Neural Networks (CNNs)**

CNNs are a class of deep learning models specifically designed for processing structured grid data, such as images. They are widely used in image recognition, object detection, and facial recognition.

**Real-Time Example: Image Classification in E-Commerce**

A fashion e-commerce website wants to automatically categorize clothing items uploaded by users. CNNs can classify images into categories like shirts, pants, and jackets based on their features.

**Key Topics and Explanations**

**1. Image Classification Example**

* Image classification is a task where an algorithm learns to assign a label to an image.
* *Example:* Classifying handwritten digits in the MNIST dataset (0-9) using CNNs.

**2. What is Convolution?**

* Convolution is a mathematical operation applied to input data using filters (kernels) to detect patterns such as edges, textures, and colors.
* *Example:* In facial recognition, convolution layers detect different facial features like eyes, nose, and mouth at different stages.

**3. Convolutional Layer Network**

* A CNN consists of multiple convolutional layers that extract spatial features from images.
* *Example:* In medical imaging, CNNs help detect tumors by learning patterns from thousands of X-ray images.

**4. Convolutional Layer**

* The first layer in a CNN that applies filters to extract features from the image.
* *Example:* Detecting edges in an image (horizontal and vertical lines) using Sobel filters.

**5. Filtering**

* Filtering helps detect important image features like textures and shapes.
* *Example:* Google Photos uses filtering techniques to identify and group similar faces in albums.

**6. ReLU Layer (Rectified Linear Unit)**

* The activation function in CNNs that introduces non-linearity.
* *Example:* Helps improve the accuracy of models in self-driving cars by distinguishing between road signs and obstacles.

**7. Pooling**

* Pooling layers reduce the spatial size of feature maps, retaining essential information while reducing computation.
* Types: Max pooling, Average pooling.
* *Example:* In Instagram’s photo recognition, pooling helps in downsampling images while preserving key features.

**8. Data Flattening**

* Converts the 2D feature maps into a 1D vector to be fed into a fully connected layer.
* *Example:* In text recognition, the final CNN layer converts character patterns into a text prediction output.

**9. Fully Connected Layer**

* Connects all neurons from previous layers to classify features extracted by CNNs.
* *Example:* A CNN model predicting whether an image contains a dog or a cat.

**10. Predicting a Cat or a Dog**

* A common CNN task where a model is trained to distinguish between different animals using labeled images.
* *Example:* Pet identification apps use CNNs to detect breeds based on pet images.

**11. Saving and Loading a Model**

* CNN models can be saved and reused for future predictions without retraining.
* *Example:* A security camera system saves its trained face recognition model to detect intruders in real-time.

**12. Face Detection using OpenCV**

* OpenCV is a computer vision library used for real-time image processing.
* *Example:* Social media platforms use OpenCV to detect faces and suggest tags in uploaded photos.

**Hands-on Practical Applications**

**1. Saving and Loading a Model**

* Saving trained CNN models for later use in different applications.
* *Example:* A smart surveillance system saves a face recognition model to identify authorized personnel.

**2. Face Detection using OpenCV**

* Detecting faces in images or live video streams using OpenCV’s pre-trained models.
* *Example:* Biometric attendance systems use OpenCV to verify employees based on facial recognition.

**Skills You Will Learn**

* Image Classification using CNN
* Face Detection using OpenCV
* Practical AI applications in real-world scenarios

## **Module 8:**

**Artificial Neural Networks (ANN)**

**What is ANN?**

Artificial Neural Networks (ANNs) are computing systems inspired by biological neural networks. They consist of layers of interconnected nodes (neurons) that process data and learn patterns.

**Key Concepts:**

* **Input Layer:** Accepts raw data inputs
* **Hidden Layers:** Extracts meaningful patterns from data
* **Output Layer:** Produces the final prediction or classification
* **Activation Functions:** Helps neurons decide whether to activate (e.g., ReLU, Sigmoid, Tanh)
* **Backpropagation:** Learning technique used to minimize errors

**Real-time Example:**

ANNs are widely used in facial recognition systems. When you unlock your phone using Face ID, an ANN processes the image and matches it with stored facial data.

**Module 9: Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM)**

**What is RNN?**

Recurrent Neural Networks (RNNs) are a type of neural network designed for sequential data. They have loops that allow them to retain information from previous steps, making them useful for tasks like speech recognition and time-series forecasting.

**Key Concepts:**

* **Sequential Processing:** Ideal for text, speech, and time-series data
* **Vanishing Gradient Problem:** RNNs struggle with long sequences
* **Applications:** Sentiment analysis, speech-to-text conversion

**What is LSTM?**

Long Short-Term Memory (LSTM) is an advanced version of RNN designed to handle long-term dependencies. LSTMs have memory cells that help them remember important features from past inputs while forgetting irrelevant details.

**Real-time Example:**

LSTMs power chatbots like ChatGPT by processing previous conversation history to generate meaningful responses.

**Module 10: Transformers**

**What are Transformers?**

Transformers are deep learning models designed to handle sequential data more efficiently than RNNs. They use self-attention mechanisms to process all input data simultaneously, improving speed and accuracy.

**Key Concepts:**

* **Self-Attention Mechanism:** Assigns different importance levels to different words in a sentence
* **Encoder-Decoder Architecture:** Used for translation and summarization
* **Applications:** Machine translation, text summarization, speech recognition

**Real-time Example:**

Google Translate uses transformer models to translate text from one language to another in real-time.