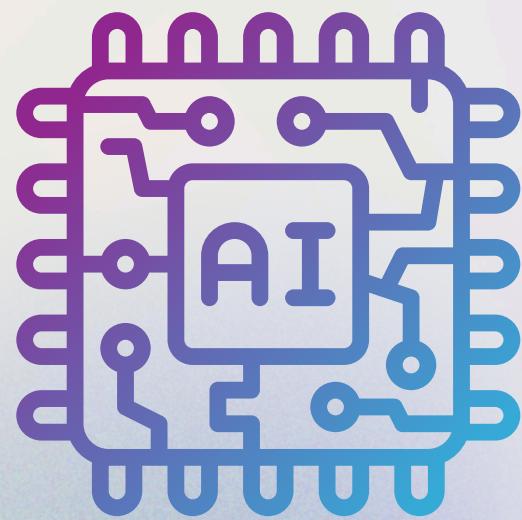




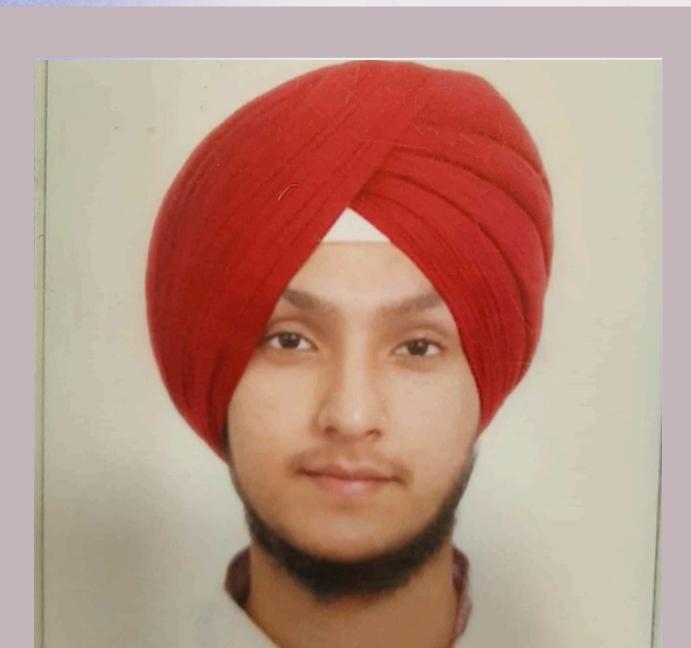
# VOICE LINGUA

An Innovative Interface for Natural Language Processing Tasks

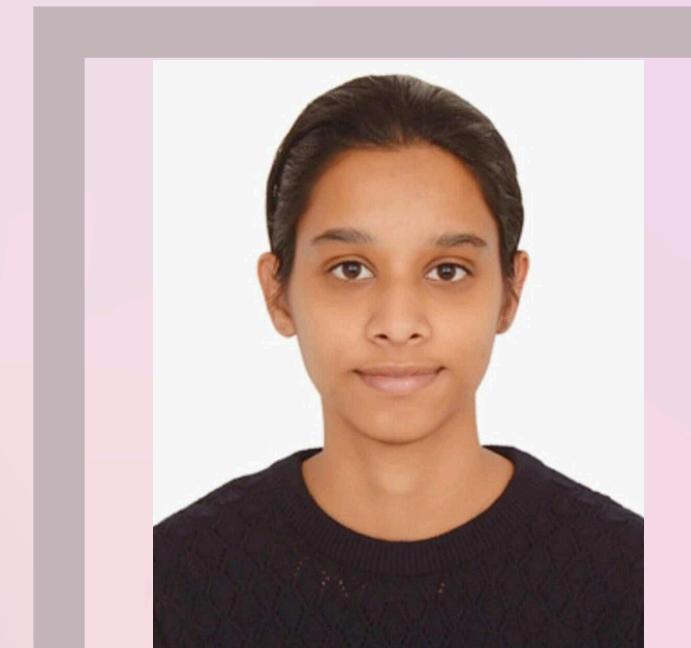




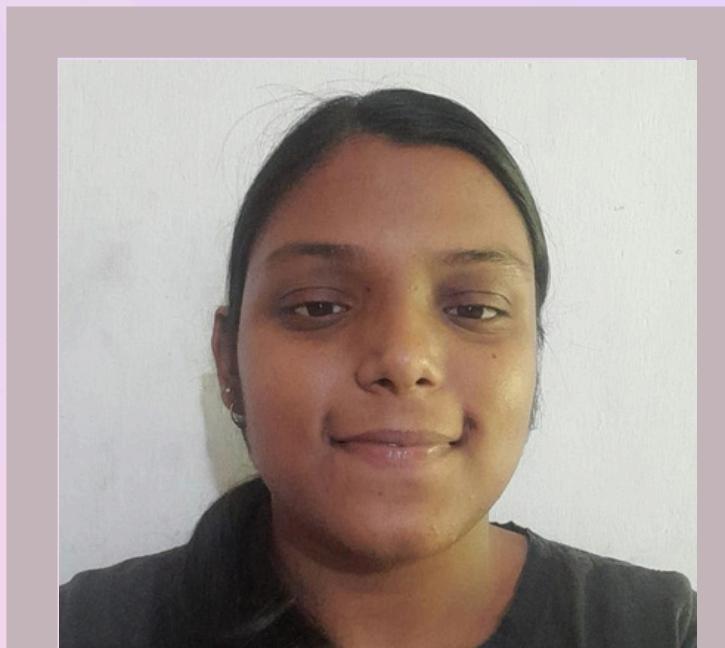
# GEN-SPARK



**HARSHDEEP  
SINGH**



**KIRANDEEP  
KAUR**



**LIZA  
KUMARI**

# INTRODUCTION

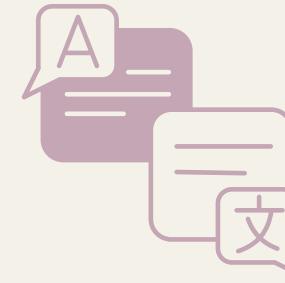
VOICE LINGUA is an innovative interface developed using Python and advanced AI technologies to provide a comprehensive suite of Natural Language Processing (NLP) tasks. It is designed to enhance the user experience with high accuracy and versatility in language processing.



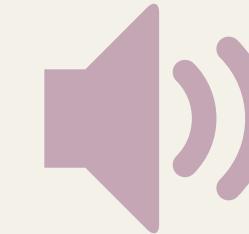
# COMPREHENSIVE NLP TASKS



Speech Recognition



Translation



Speech Generation



Audio Extraction



Summarization

# SPEECH RECOGNITION

- Converts speech into text in multiple languages.
- Support 50+ languages.
- Given by openai/whisper-large-v3 model

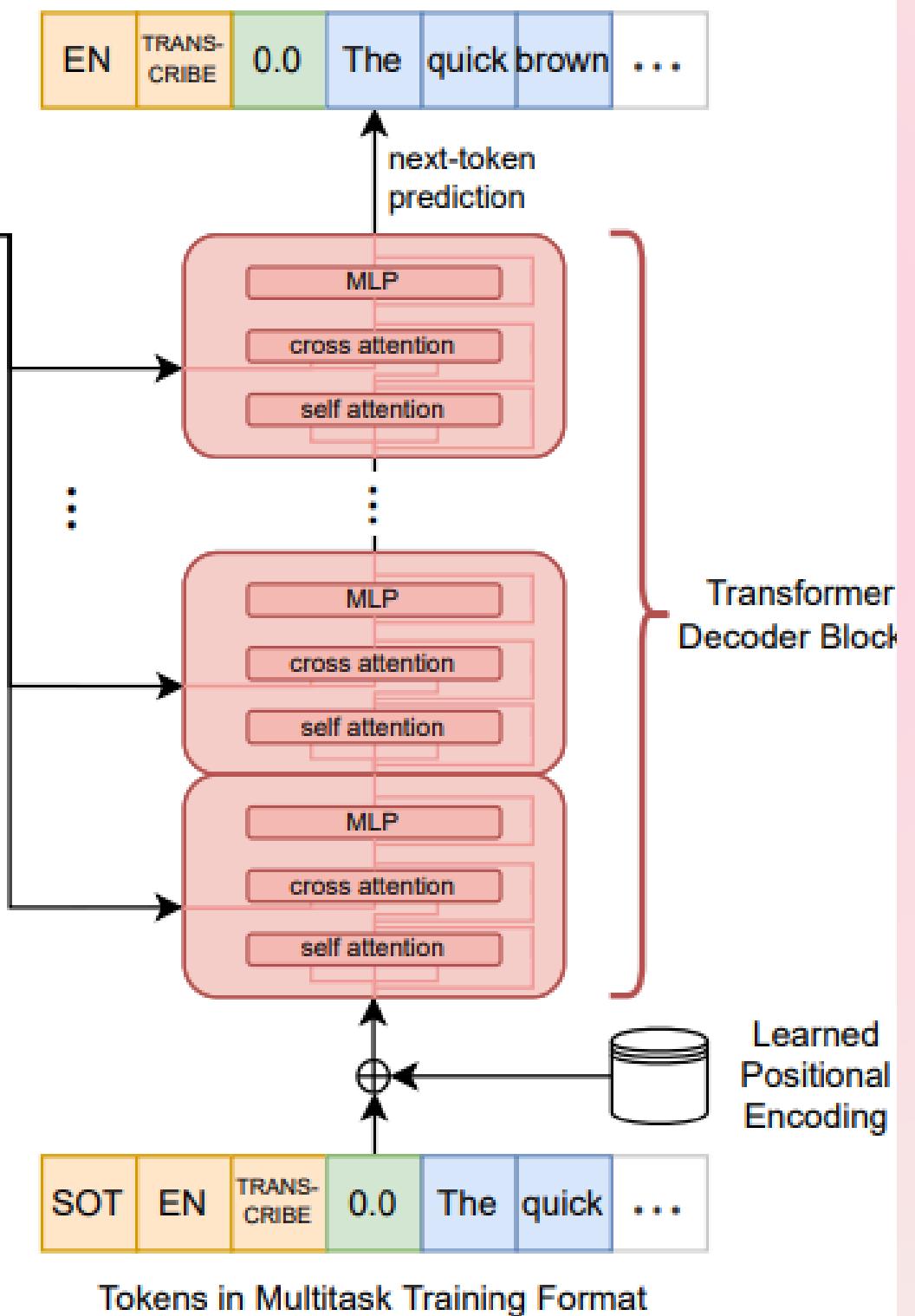
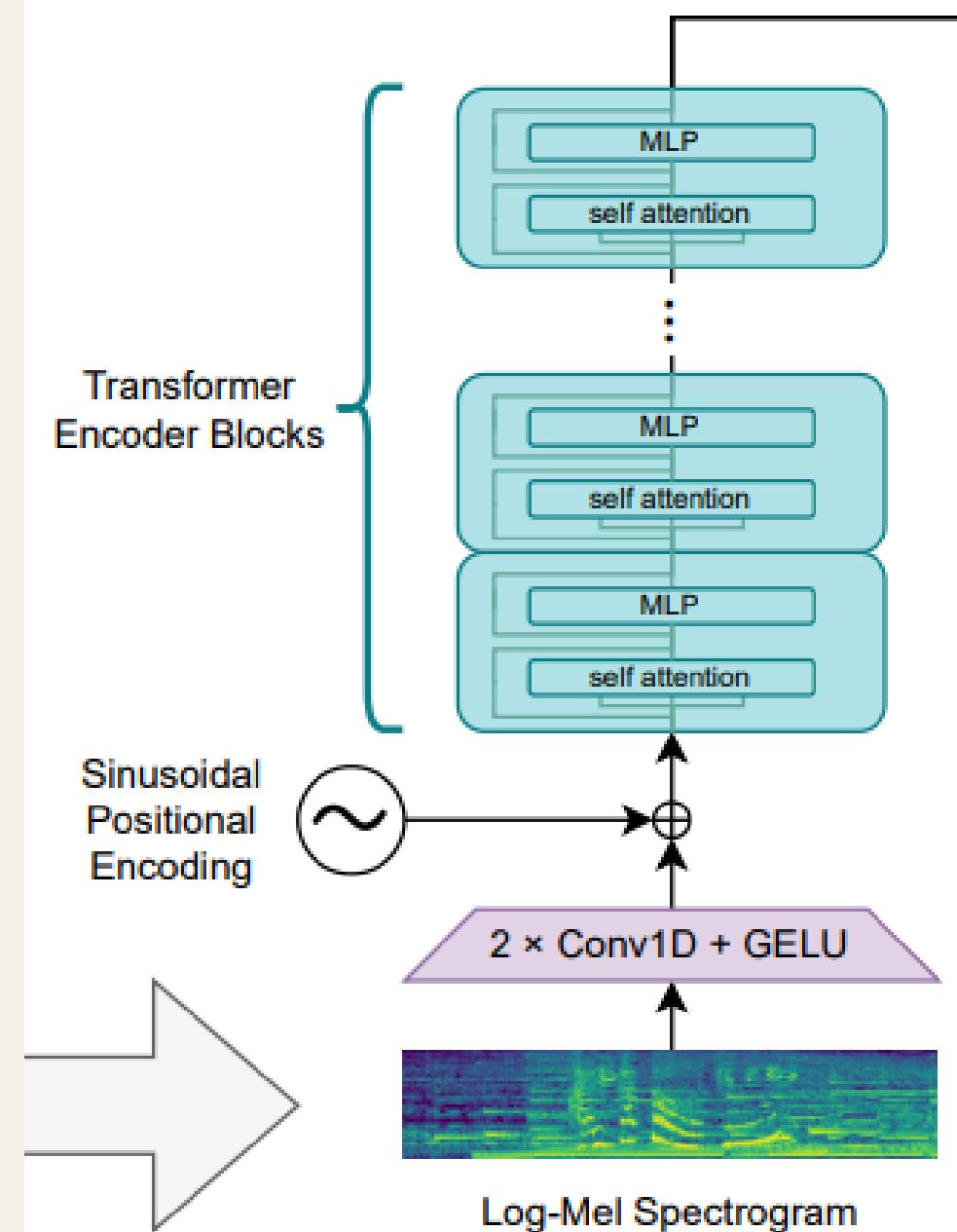
The screenshot shows the VOICE-LINGUA web application. On the left, there's a circular logo with a microphone icon and the text "VOICE LINGUA". Below it, the text "VOICE-LINGUA" is displayed. A sidebar on the left lists options: "Select an option:" followed by "Speech Recognition" (which is checked), "Translation", "Speech Generation", "Audio Extraction", and "Summarization". The main content area has a title "Speech Recognition" and a sub-section "Upload an audio file to transcribe:". It features a "Drag and drop file here" button with a 200MB limit for WAV, MP3, M4A, MPEG, MPG files, and a "Browse files" button.

The screenshot shows a "Speech Recognition" tool. At the top, it says "Upload an audio file to transcribe:". Below that is a "Drag and drop file here" field with a 200MB limit for WAV, MP3, M4A, MPEG, MPG files, and a "Browse files" button. An uploaded file "audio file.mpeg 1.0MB" is listed with a delete "X" button. Below the file list is a progress bar showing "0:00 / 0:26". To the right of the progress bar are volume and settings icons. Underneath the file list, there's a "Transcription:" section with a descriptive paragraph about education's role in personal and social growth. At the bottom, it says "Language Code:" followed by "eng\_Latn".

# WORKFLOW

- 1. Audio Preprocessing:** Convert audio signals into Mel spectrogram frames.
- 2. Feature Extraction:** Extract important features from the audio frames.
- 3. Positional Encoding:** Add positional information to the frames.
- 4. Transformer Encoder:** Process frames to capture long-range dependencies.
- 5. Transformer Decoder:** Decode frames to generate the text token-by-token.
- 6. Output Generation:** Produce the final textual output

## Sequence-to-sequence learning



# TRANSLATION

- Allows you to translate effectively between source and target languages.
- 50+ languages are supported.
- Powered by NLLB-200 model

**VOICE-LINGUA**

Select an option:

- Speech Recognition
- Translation
- Speech Generation
- Audio Extraction
- Summarization

**Translation**

Translate text from one language to another:

Enter text to translate:

Select source language:

en

Select target language:

en

Translate

Translate text from one language to another:

Enter text to translate:

Everything we see around us constitutes nature, including the sun, the moon, trees, flowers, fruits, human beings, birds, animals, etc. In nature, everyone depends on one another to keep the ecosystem healthy. For survival, every creature is interrelated and reliant on one another. Humans, for example, rely on nature for their survival, and nature provides us with oxygen, food, water,

Select source language:

en

Select target language:

hi

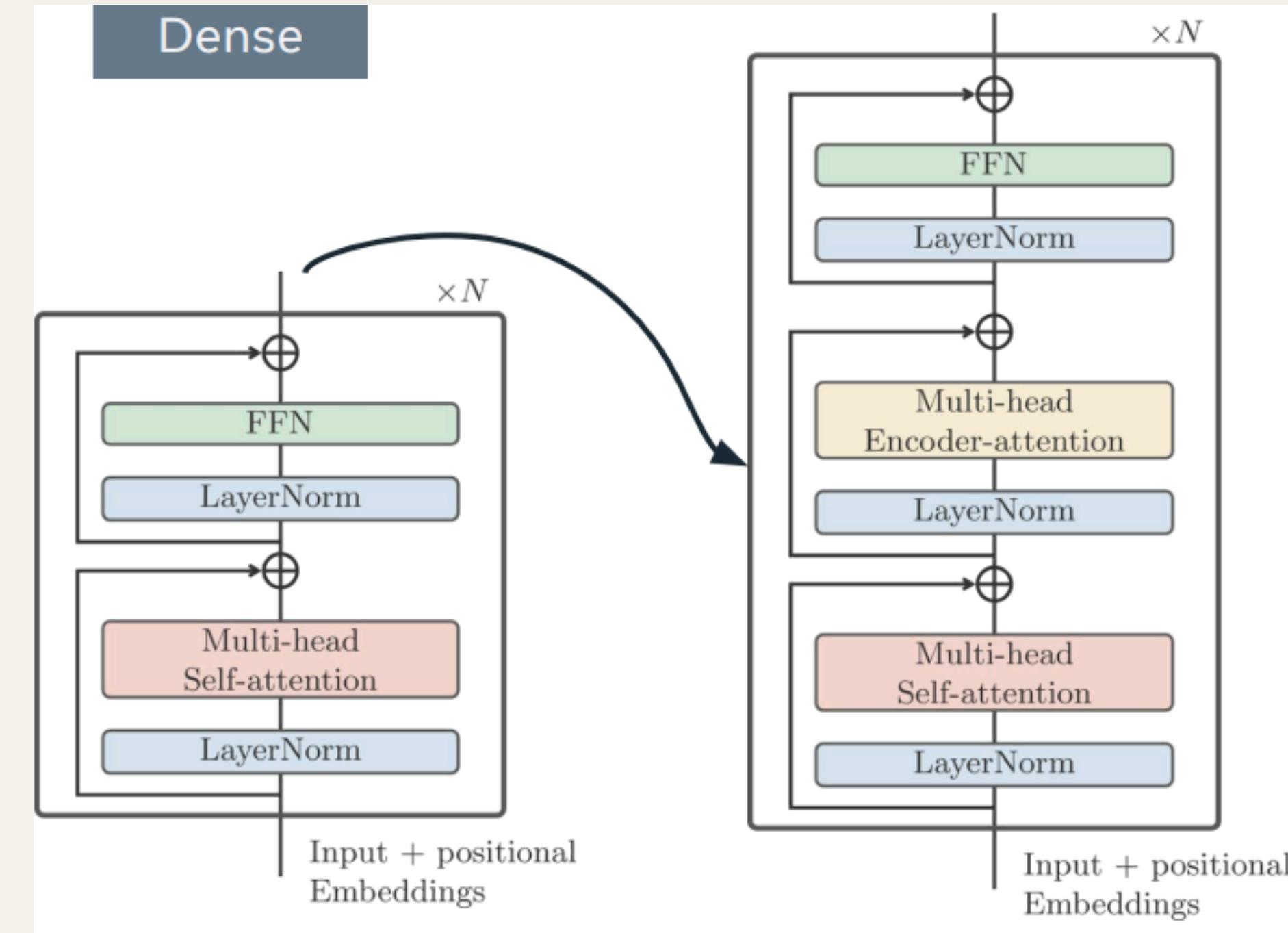
Translate

**Translated Text:**

हमारे चारों ओर जो कुछ भी हम देखते हैं वह प्रकृति का निर्माण करता है, जिसमें सूर्य, चंद्रमा, पेड़, फूल, फल, मनुष्य, पक्षी, जानवर आदि शामिल हैं। प्रकृति में, पारिस्थितिकी तत्र को स्वस्य रखने के लिए सभी एक दूसरे पर निर्भर हैं। अस्तित्व के लिए, प्रत्येक प्राणी परस्पर संबंधित है और एक दूसरे पर निर्भर है। उदाहरण के लिए, मनुष्य अपने अस्तित्व के लिए प्रकृति पर निर्भर है, और प्रकृति हमें ऑक्सीजन, भोजन, पानी, आश्रय, दवाएं और कपड़े प्रदान करती है, अन्य चीजों के अलावा।

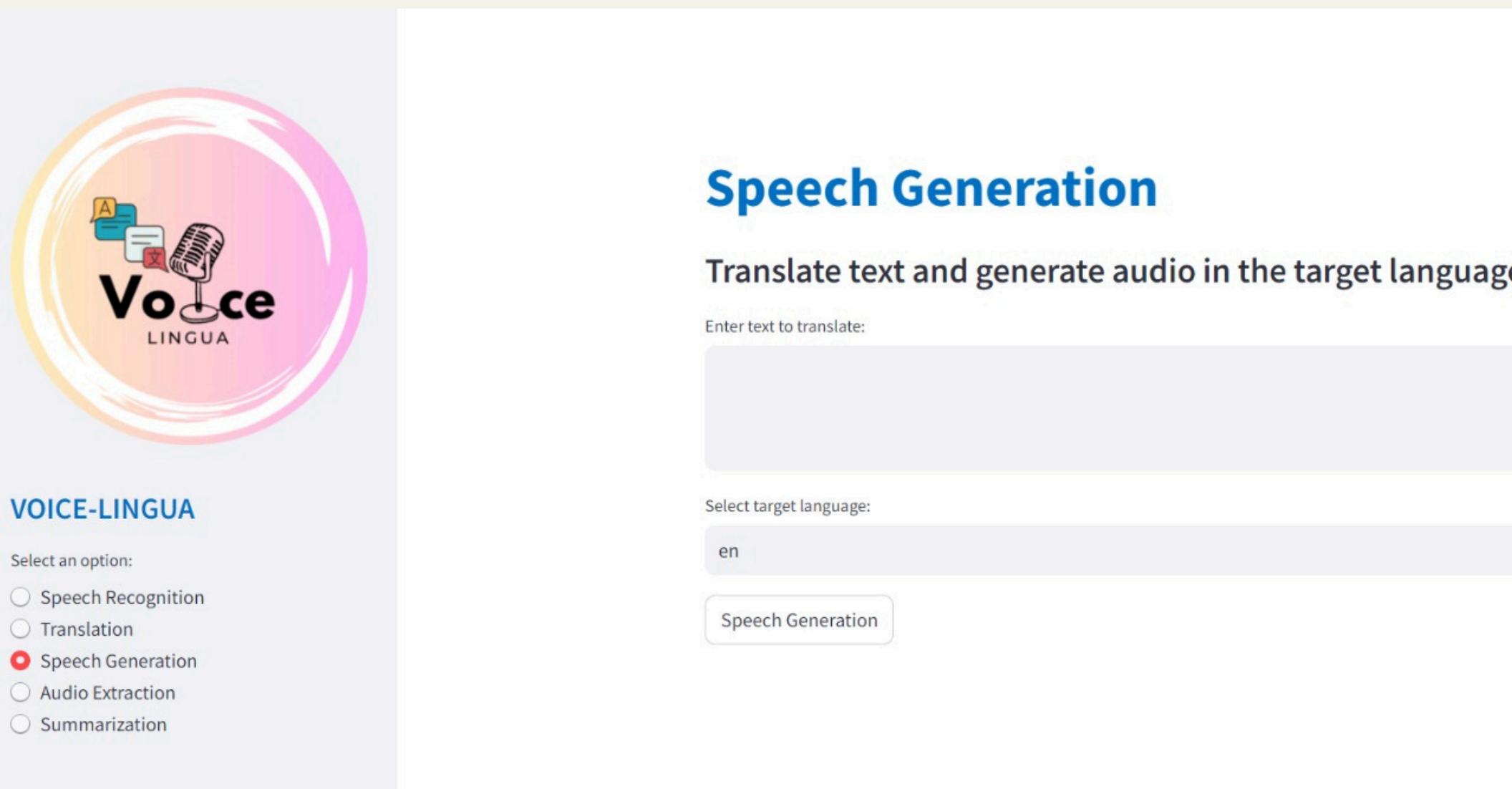
# WORKFLOW

- 1. Tokenization:** Convert input text into tokens.
- 2. Positional Encoding:** Add positional information to the tokens.
- 3. Transformer Encoder:** Process tokens to capture contextual relationships.
- 4. Transformer Decoder:** Decode tokens sequentially to generate the translation.
- 5. Output Generation:** Detokenize the generated tokens to produce the final translated text.



# SPEECH GENERATION

- Converts text into natural-sounding speech with remarkable accuracy and clarity.
- Users can input text in any supported language and receive high-quality speech output.
- Supported by the gTTS library.



The screenshot shows the VOICE-LINGUA web application. On the left, there's a circular logo with a microphone icon and the text "VOICE LINGUA". Below it, a sidebar lists options: "Select an option:" followed by "Speech Recognition" (unchecked), "Translation" (unchecked), "Speech Generation" (checked), "Audio Extraction" (unchecked), and "Summarization" (unchecked). The main content area has a title "Speech Generation" and a subtitle "Translate text and generate audio in the target language:". It includes fields for "Enter text to translate:" (empty), "Select target language:" (set to "en"), and a "Speech Generation" button. To the right, a larger window titled "Speech Generation" shows the same interface but with "fr" selected as the target language. It displays the translated text "Everything we see around us constitutes nature, including the sun, the moon, trees, flowers, fruits, human beings, birds, animals, etc. In nature, everyone depends on one another to keep the ecosystem healthy. For survival, every creature is interrelated and reliant on one another. Humans, for example, rely on nature for their survival, and nature provides us with oxygen, food, water," and a "Speech Generation" button. A green notification bar at the bottom says "Audio file generated successfully!". Below it is a media player showing a progress bar from 0:00 to 0:42.

# WORKFLOW

**1. Text Translation:** The code uses the GoogleTranslator from the googletrans library to translate the input text.

The translate method is called on the translator object, passing the input text as the argument.

The translated text is stored in the translated text variable.

The input text is translated into the target language using the Google Translator API. The source language is automatically detected, and the target language is specified by the user.

**2. Audio Generation:** The translated text is then used to generate an audio file using the Google Text-to-Speech (gTTS) API. The target language is specified to ensure the correct pronunciation and accent.

**3. Audio File Saving:** The generated audio file is saved with a specified filename.

**4. Error Handling:** If the target language is not valid, an error message is displayed.

**5. Function Call:** The `translate and generate audio` function is called with the input text, target language, and filename as arguments.

**6. Output:** The generated audio file is saved and can be played back.

# AUDIO EXTRACTION

- Converts video files into high-quality audio.
- Supports various video file types , ensuring compatibility and ease of use.



**VOICE-LINGUA**

Select an option:

- Speech Recognition
- Translation
- Speech Generation
- Audio Extraction
- Summarization

## Audio Extraction

Extract audio from a video file:

Enter the path to the video file:

Extract Audio

## Audio Extraction

Extract audio from a video file:

Enter the path to the video file:

Extract Audio

Audio extracted successfully!

Download the extracted audio file

Audio file not downloaded.

# WORKFLOW

## **Import the MoviePy Module:**

The code starts by importing the `moviepy.editor` module, which provides the necessary functions and classes for video editing.

## **Load the Video File:**

The `VideoFileClip` class from the `moviepy.editor` module is used to load the video file.

The video file path is passed as an argument to the `VideoFileClip` function, which creates a `VideoFileClip` object representing the video.

## **Extract the Audio:**

The `audio` attribute of the `VideoFileClip` object is accessed to extract the audio from the video.

This `audio` attribute returns an `AudioFileClip` object, which represents the audio component of the video.

## **Write the Audio to a File:**

The `write_audiofile` method of the `AudioFileClip` object is used to save the extracted audio to a file.

The desired filename (e.g., "output.mp3") is passed as an argument to the `write_audiofile` method.

# SUMMARIZATION

- Provides precise and coherent summaries of lengthy texts, reducing the content into customizable lengths.
- Users can enter any type of writing and get brief summaries that preserve important details.
- Model used : facebook/bart-large-cnn



**VOICE-LINGUA**

Select an option:

- Speech Recognition
- Translation
- Speech Generation
- Audio Extraction
- Summarization

## Text Summarizer

Enter the text to summarize:

Number of words in the input text: 234

Choose the minimum summary length 0

Number of words in the input text: 234

Choose the maximum summary length 234

**Summary:**

Human activities have an impact on nature, and as a result, the quality of the environment is deteriorating . The oxygen produced by a single fully-grown tree is enough to supply ten people . The benefits of sunlight and fresh air to our health cannot be overstated . Earthquakes, volcanic eruptions, floods, and cyclones are examples of natural calamities . People prefer to raise their children in the countryside in the U.S. Their primary goal is to introduce their children to the benefits of natural beauty .

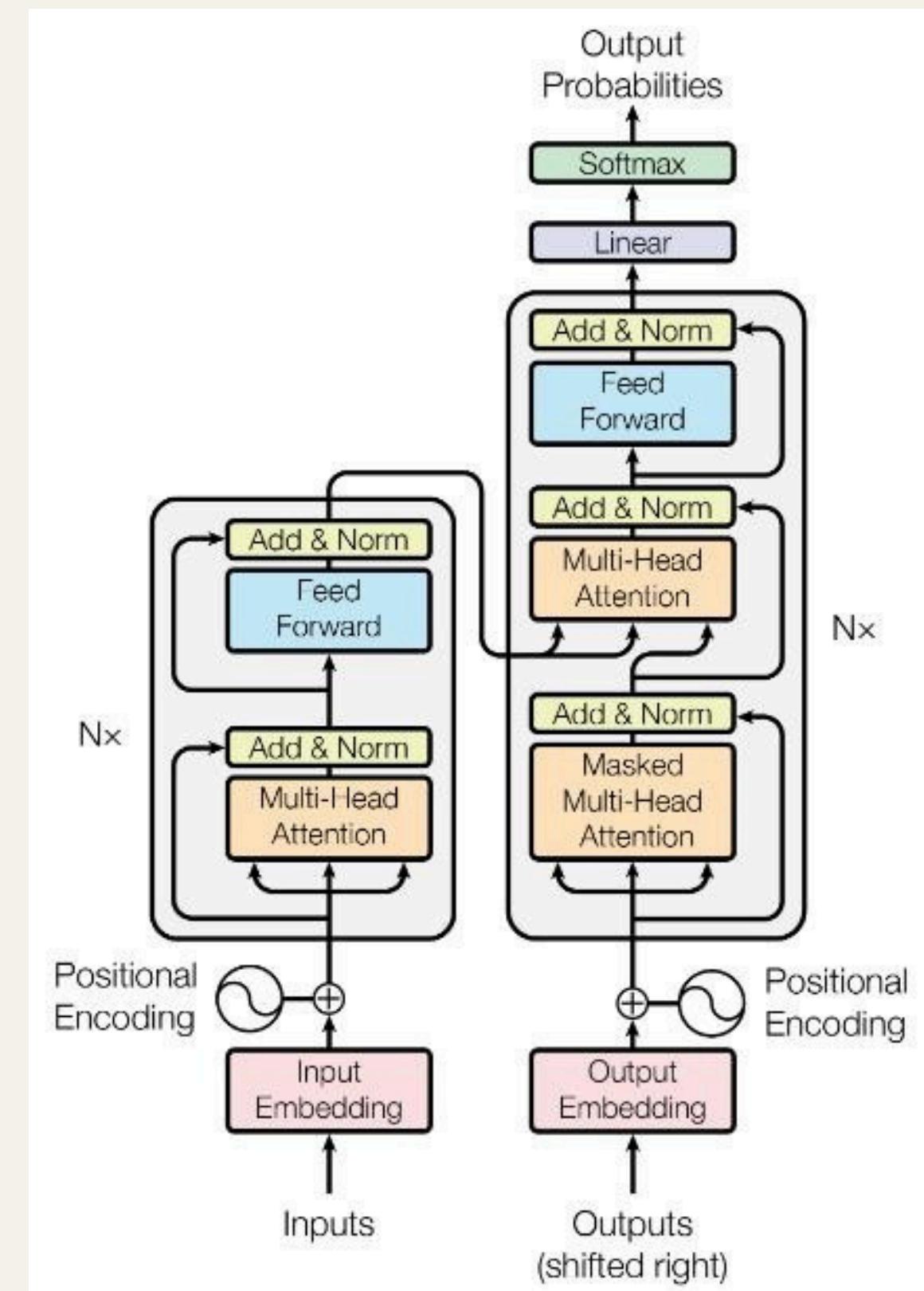
# WORKFLOW

## Summary Generation:

1. **Input Encoding:** the input text is tokenized and the encoder generates contextualized representation.
2. **Denoising Objective:** Training for robust representations with masked/corrupted inputs
3. **Decoding:** Generates summaries by predicting tokens based on encoded context.

## Sentence and Word Selection:

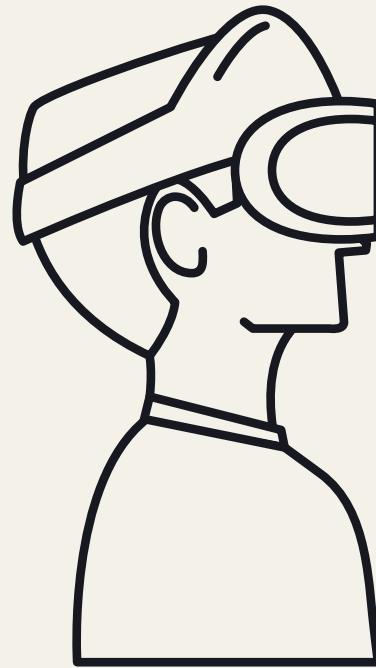
1. **Attention-Mechanism:** Self-attention and cross-attention focus on relevant parts of the input.
2. **Positional Encoding:** Maintains token order for coherent summaries.
3. **Beam Search:** Tracks multiple summaries and selects the best one.
4. **Length Penalty:** Ensures concise and to-the-point summaries.



# USE CASE

**VOICE LINGUA is applicable in various fields, including:**

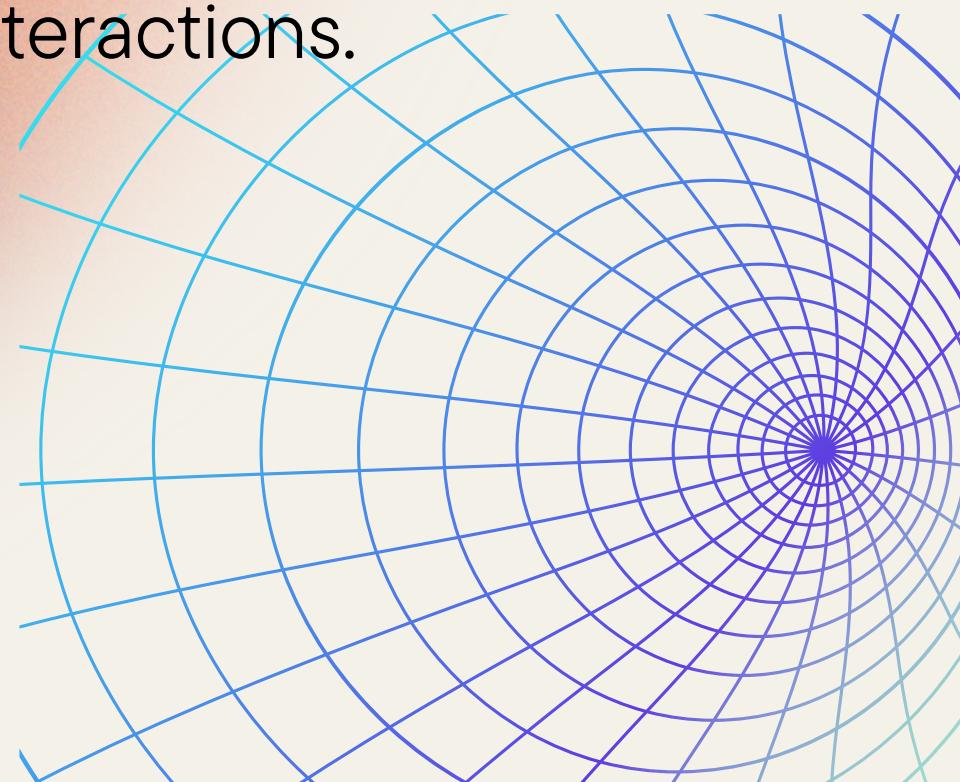
- **Medical:** Improving healthcare communication by transcribing doctor-patient interactions, translating medical documents, summarizing patient notes, and generating audio content for patient.
- **Education:** Facilitating transcription, translation, and summarization of study materials for students and educators.
- **Business:** Enhancing productivity with accurate meeting transcriptions, multilingual communication, and content summarization.
- **Content Creation:** Assisting creators with generating voiceovers, translating content for a global audience, and extracting audio from videos for podcasts.
- **News and Multimedia:** Supporting journalists and producers in transcribing interviews, translating reports, generating multilingual news segments, and repurposing video content.

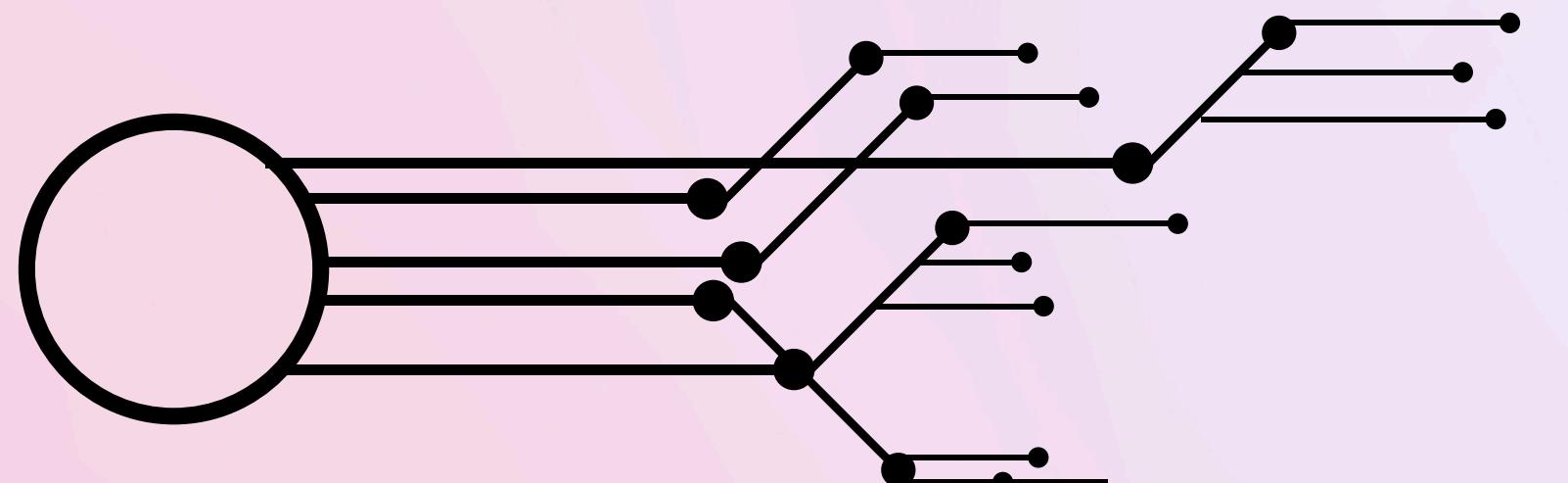


# FUTURE WORK

- 1. Improve Naturalness and Accuracy:** Speech Recognition: Improve speech recognition accuracy for a variety of accents and dialects, as well as in loud settings.
- 2. Expansion of Language Support:** Increase the number of supported languages and dialects, focusing on underrepresented languages to make the system more inclusive.
- 3. Real-time Processing:** Optimize the system for real-time processing, reducing latency in speech recognition, translation, and generation to support live interactions.

(i)





# THANK-YOU

