International Journal of Electrical, Electronics and Computers
Vol-9, Issue-5 | Sep-Oct, 2024
Available: https://aipublications.com/ijeec/
Peer-Reviewed Journal

Tackling the Problem of Multilingualism in Voice Assistants

Soham Sabharwal[1], Rohan Sahni[2]

[1]Kunskapsskolan International School, India
Email: sohamsabharwal2007@gmail.com
[2]IIT Madras, India
Email: 23f2004610@ds.study.iitm.ac.in

Received: 23 Aug 2024; Accepted: 22 Sep 2024; Date of Publication: 01 Oct 2024

Abstract**—** *Voice assistants like Alexa and Siri have become increasingly advanced due to improvements in AI and language processing models like GPT and Gemini. However, these systems often perform poorly with less commonly spoken languages, such as many Indian languages, creating a significant accessibility gap. This paper addresses the problem of multilingualism in voice assistants, with a focus on languages like Hindi, Punjabi, and Bengali. We examine the evolution of voice assistants and highlight the major technical challenges they face, including speech recognition, language processing, and response generation in low-resource languages. To overcome these barriers, we propose a novel framework that combines different AI models to enhance multilingual support. Our approach offers a potential solution to make voice assistants more inclusive and accessible for speakers of underrepresented languages. By broadening language support, this research has the potential to extend the benefits of AI to a much wider audience.*

Keywords**— *Multilingualism, Voice assistants, AI, Transformers, Large Language Models, Natural Language Understanding***

I. INTRODUCTION

Imagine being able to talk to your phone in any language and have it understand you perfectly. That's the dream of multilingual voice assistants, and it's what our paper is all about. Artificial Intelligence (AI) has come a long way since it was first introduced in the 1950s. One of the coolest things AI can do now is understand and generate human language. This ability has led to the creation of chat-based assistants like GPT and voice assistants like Alexa and Siri. While these assistants work great in English, they often struggle with less common languages [1][2]. This is a big problem because it means many people around the world can't use this awesome technology. Our paper focuses on solving this issue, especially for Indian languages like Punjabi and Bengali. Here's a quick rundown of what you'll find in our paper:

- We look at how voice assistants and voice cloning tech have evolved over time.
- We dive into the cool AI stuff that makes language processing possible, like neural networks and transformer models.
- We explore the key parts of a multilingual voice assistant, from speech recognition to generating responses.
- We discuss the tricky challenges of building assistants that work in many languages.
- Finally, we propose a new way to build these assistants using a mix of different AI models.

By the end of this paper, you'll have a good understanding of how multilingual voice assistants work and the exciting progress being made in this field.

II. BACKGROUND

Artificial Intelligence (AI) has made significant progress since its introduction in the 1950s. Over the decades, AI

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

has evolved from simple rule-based systems to complex machine learning models capable of understanding and generating human language. This advancement has led to the development of sophisticated chat-based assistants like GPT and voice assistants such as Alexa and Siri, which have become increasingly prevalent in our daily lives [4]. The journey of AI assistance began with rule-based engines, which later relied on labeled data to make predictions within specific contexts. However, these early systems lacked creativity due to their reliance on manual feature engineering and predefined rules. A major breakthrough came in 2022 with the introduction of generative AI. This new approach allowed AI models to learn patterns and structures from data, enabling them to generate new content and explore creative solutions. The shift from traditional encoder-decoder architectures to attention-based models has transformed the landscape of generative AI, leading to the emergence of more advanced chat models and improvements in large language models.

2.1. Evolution of Language Models

The evolution of chat assistants can be traced back to early systems like Jabberwacky, ELIZA, and Smarter Child, which laid the groundwork for natural language processing. As the field progressed, breakthroughs in machine learning and neural network architectures enabled the development of more advanced assistants like Siri, Alexa, and Google Assistant. The introduction of transformer-based architectures and attention mechanisms, exemplified by models like OpenAI's GPT and Google's BERT, revolutionized conversational AI.

2.2. Voice assistants-how they started

Voice assistants, a prominent application of this technology, have their history. Siri, developed in 2010 and later acquired by Apple, was one of the first widely adopted voice assistants. It allowed users to perform tasks, answer questions, and interact with their devices using voice commands. Amazon's Alexa, introduced in 2014, further expanded the capabilities of voice assistants, particularly in the realm of smart home integration.

2.3. Tech behind Voice Assistants

The technology behind voice assistants is complex and multifaceted. Siri, for instance, was created by the Stanford Research Institute and funded by DARPA. Its speech recognition engine was provided by Nuance Communications. Modern voice assistants rely on a complex pipeline of technologies, primarily involving automatic speech recognition (ASR), natural language understanding (NLU), dialogue management, and text-to-speech (TTS) synthesis. These systems are built on advanced artificial intelligence and machine learning techniques, including deep neural networks and transformer-based models. They utilize large-scale data processing and cloud computing infrastructures, with a growing trend towards edge computing for certain on-device functionalities. Natural language processing (NLP) techniques are crucial, enabling these assistants to interpret context, intent, and linguistic nuances. This paper will delve into these technologies in subsequent sections, examining how they integrate to produce responsive and intelligent voice interfaces. The synergy of these components allows voice assistants to process and understand spoken language, formulate appropriate responses, and interact with users in an increasingly natural manner.

2.4. Voice Cloning

A significant recent development in this field is voice cloning technology. This involves creating a synthetic copy of a human voice, which has important implications for multilingual voice assistants. Voice cloning could potentially allow for the generation of natural-sounding speech in multiple languages without the need for extensive voice recordings from native speakers. The process typically involves data collection, feature extraction, model training, and voice synthesis.

The flowchart in Figure 1 shows the four main steps involved in voice cloning:

1. Data Collection, where high-quality voice samples are collected from the target speaker.
2. Feature Extraction, which involves analyzing and extracting key voice characteristics such as pitch, timbre, and rhythm.
3. Model Training, where machine learning models are trained on the extracted features to learn the unique qualities of the voice.
4. Voice Synthesis, where the trained model generates new speech in the cloned voice.

In the context of multilingual voice assistants, voice cloning offers several advantages:

1. Language Expansion: It allows voice assistants to "speak" in languages for which they don't have native recordings, enabling rapid expansion into new linguistic markets.
2. Personalization: Users could potentially have their voice assistants speak in voices of their

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

choosing, including their own or those of loved ones.

3. Consistency: It ensures a consistent voice across different languages, maintaining brand identity for commercial voice assistants.
4. Resource Efficiency: It reduces the need for extensive voice recording sessions with native speakers of every target language.

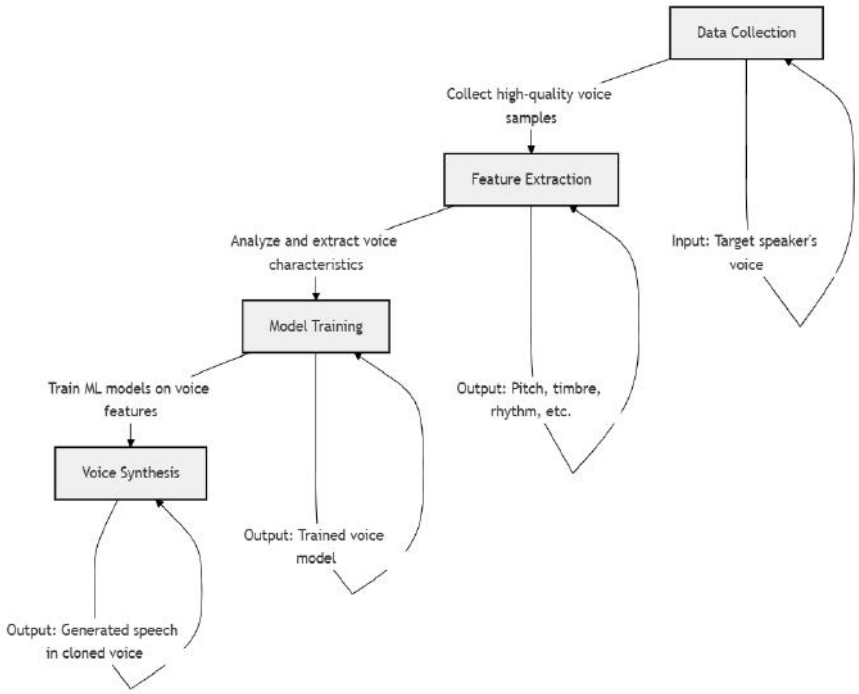

Fig. 1: Flowchart of the voice cloning process.

Voice cloning offers several advantages for multilingual voice assistants, including rapid language expansion, personalization options, brand consistency across languages, and resource efficiency. However, it also presents challenges in maintaining speech naturalness across different languages and raises ethical considerations regarding consent and potential misuse. Companies like ElevenLabs and others mentioned in this paper are at the forefront of developing voice cloning technologies [3]. These advancements have the potential to revolutionize how multilingual voice assistants sound and interact across different languages, paving the way for more inclusive and globally accessible AI assistants.

III. LITERATURE REVIEW

The following section provides a comprehensive overview of key advancements in language processing technologies, with a focus on neural networks, transformer models, and multilingual language models.

3.1. Neural Networks in Language Processing:

Neural networks are powerful tools for analyzing complex data like images, videos, audio, and text. Different types of neural networks work best for different kinds of data. For example, convolutional neural networks (CNNs) are great for image processing because they mimic how our brains process visual information. However, CNNs aren't the best choice for language tasks because they mainly focus on local patterns and struggle with understanding long-range connections between words in a sentence.

Language modeling has come a long way in the past decade. It started with simple techniques like Word2Vec and n-grams in 2013, which helped represent words as vectors. This was a big deal because it allowed computers to understand the meaning and relationships between words based on their context. Then, in 2014, models like RNNs and LSTMs got better at analyzing and generating sequences of words, which was super helpful for tasks like translation. Around 2015, attention mechanisms were introduced. These allowed models to focus on important parts of input sequences, making

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

them better at understanding longer sentences and improving tasks like translation and summarization. The real game-changer came in 2017 with the paper "Attention Is All You Need," which introduced the Transformer architecture [5]. Transformers solved two big problems that RNNs had:

1. RNNs struggled with long texts because they tended to forget the earlier parts.
2. RNNs processed data sequentially, which made them slow to train.

Transformers can process data in parallel, making them much faster and more efficient. They're also really good at handling massive datasets, which is crucial for language tasks.

3.2. Transformer Architecture:

The Transformer architecture has three main parts:

3.2.1. Attention Mechanism: This was first used to improve machine translation. It lets the model focus on different parts of the input while creating each part of the output. This was a big step towards Transformers because it showed how powerful it could be to directly model relationships between different parts of a sequence.

3.2.2. Self-Attention: This is a special type of attention used in Transformers. It allows each word in a sentence to look at all the other words in the same sentence. This helps the model understand things like synonyms and grammar rules. For example, in the sentence "The cat, which was very hungry, chased the mouse," self-attention helps the Transformer know that "cat" and "chased" are related, even though there are other words between them.

3.2.3. Positional Encoding: Since Transformers don't process words in order, they need another way to understand word order. Positional encoding adds a number to each word to show its position in the sentence. The model learns how important these position numbers are during training, which helps it understand the order of words from the data.

Transformers have become super important in natural language processing. They're used in different ways depending on the task:

- Encoder-Decoder: Models like mBART use this setup for tasks like translation. The encoder processes the input text, and the decoder generates the translated output.
- Encoder-only: Models like mBERT use just the encoder part. They're great for tasks that need to understand input text, like sentiment analysis or identifying named entities in a sentence.
- Decoder-only: Models like XGLM use just the decoder part. They're good at generating text, like completing a sentence you start.

Each of these setups has its strengths, and researchers choose the best one depending on what they're trying to do with language.

3.3. Multilingual Language Models

BERT, introduced by Google in 2018, revolutionized this field by introducing bidirectional training of transformers, allowing the model to understand the context from both the left and right sides of a word. It is a pre-trained model in NLP that uses a "masked language model" (MLM) pre-training objective. It randomly masks some tokens in the input and aims to predict their original words, capturing the context and relationships between words. Traditional language models process texts in an unidirectional manner however human language understanding relies on context from both directions. For example, the sentence “The bank by the river is closed”, requires context from both directions to understand the meaning of bank. BERT takes into account not only the words that come before it but also those that follow. In the context of multilingualism, the bidirectional approach allows models like BERT to create a robust and versatile representation of language, understand cross-lingual similarities and differences, and perform well in various NLP tasks like question answering and named entity recognition.

A more general solution is to produce multilingual models that have been pre-trained on a mixture of many languages. Popular models of this type are mBERT (Devlin, 2018), mBART (Liu et al., 2020a), and XLM-R (Conneau et al., 2020), which are multilingual variants of BERT (Devlin et al., 2019), BART (Lewis et al., 2020b), and RoBERTa (Liu et al., 2019), respectively. The remarkable aspect of these models is their capacity to learn a language during pre-training and enhance their expertise for specific tasks during fine-tuning.

Another significant advancement involves teaching a model to comprehend multiple languages, leading to the concept of multilingual LLMs. While monolingual models focus on understanding patterns within a single language, multilingual LLMs simultaneously learn from multiple languages. This is accomplished by exposing these models to data from various languages during the

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

pre-training phase. Furthermore, variations in the architectures of multilingual LLMs contribute to their strengths in certain tasks while potentially limiting their effectiveness in others. One common approach is to pre-train the base model on a diverse range of monolingual corpora from different languages. This allows the model to learn representations of words and sentences in multiple languages.

Additionally, techniques such as cross-lingual pre-training and fine-tuning, as well as the use of language-specific modules, are employed to adapt the base model to handle linguistic diversity and nuances across different languages. Moreover, adapter-based approaches and training new embedding layers with corresponding target-language tokenizers are utilized to extend monolingual models to new languages and improve cross-lingual capabilities. These techniques aim to enhance the performance of multilingual models and address the challenges associated with linguistic diversity and data sparsity. Now let's have a brief overview of the most prominent multilingual models also including those specifically designed for Indian languages:

3.3.1. mBert

The multilingual version of BERT (mBert) is an encoder-only transformer model that can perform cross-lingual transfer learning which means the model trained in one language can be applied to tasks in other languages without additional training. It is trained on Wikipedia data of 104 languages which offers broad language coverage and the ability to perform zero-shot cross-lingual transfer that can be very useful for low-resource languages. However, due to its large size, it can be computationally expensive and less suitable for real-time and low-latency applications. Furthermore, BERT's multilingual capabilities, while impressive, might not be equally effective across all languages. The model's performance varies across different languages, and it may not be as proficient in low-resource languages due to the limited data availability. mBERT's fine-tuning process can be complex and requires careful optimization for specific tasks in different languages and demands expertise and computational resources, which limit certain practical applications.

3.3.2. MT5

T5, which stands for "Text-to-Text Transfer Transformer," is a groundbreaking language model that revolutionized the approach to NLP tasks. It uses a unified "text-to-text" format, allowing it to handle various NLP challenges with a single set of hyperparameters. This versatility makes fine-tuning across different tasks more efficient. T5 employs an encoder-decoder Transformer architecture and is pre-trained using a "span-corruption" objective. In this pre-training process, the model learns by reconstructing masked spans of input tokens, enhancing its understanding of language structure. T5 comes in different sizes, ranging from 60 million to a massive 11 billion parameters, and was pre-trained on about 1 trillion tokens from the C4 dataset - a vast collection of English text from Common Crawl, totaling around 750GB. The model's strengths lie in its unified approach, effective design, and scalability, making it adaptable to a wide array of NLP tasks.

mT5, the multilingual version of T5, takes these capabilities to the next level by extending them to multiple languages. It maintains the unified "text-to-text" format, allowing consistent handling of various NLP tasks across different languages. This multilingual model was pre-trained on a dataset derived from Common Crawl, encompassing an impressive 101 languages. This extensive language coverage significantly enhances mT5's ability to understand and generate text in numerous languages. Like its predecessor, mT5 utilizes an encoder-decoder Transformer architecture, which is particularly effective for processing and generating text in a multilingual context. After its initial pre-training phase, mT5 can be fine-tuned for specific multilingual tasks, aiming to achieve state-of-the-art performance on various multilingual NLP benchmarks. This adaptability makes mT5 a powerful tool for tackling complex language tasks across a diverse range of linguistic environments.

3.3.3. mBART

We are well aware that mBERT is used for tasks such as classifying text, sentiment analysis, or recognizing named entities because it focuses on understanding the input text. On the other hand, for text generation tasks we have mBART which is a sequence-to-sequence multilingual model. It learns to fix corrupted texts from a process called "denoising auto-encoding". mBART corrupts input texts by masking phrases and permuting sentences. The model is then trained to reconstruct the original text from this corrupted version. The key innovation of mBART is its ability to pre-train a complete Seq2Seq model by denoising full texts in multiple languages, unlike previous approaches that only focused on either the encoder, decoder, or parts of the text. This pre-training allows the model to learn a robust understanding of the structure and semantics of multiple languages, making it highly effective for

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

downstream tasks like translation. mBART can help learn across different languages, even those not used during training. This means it can adapt well to new languages. mBART works well for both short sentences and longer documents. It shows big improvements whether the translation task is supervised (with specific guidance) or unsupervised (with less specific guidance).

3.3.4. BLOOM and XLM-R

Similar to above mentioned multilingual models, we have BLOOM and XLM-R that have been developed to access various language processing needs across different linguistic groups. BLOOM and XLM-R are advanced multilingual language models designed to handle a wide range of natural language processing tasks across multiple languages. BLOOM, developed through a large-scale collaboration involving hundreds of researchers, is notable for its 176 billion parameters, supporting 46 natural languages and 13 programming languages, and excels in both zero-shot and few-shot learning scenarios. XLM-R, created by Facebook AI, supports 100 languages and uses the Transformer architecture, pre-trained on the massive CC100 dataset. It has various model sizes, up to 10.7 billion parameters, and is particularly strong in handling low-resource languages, leveraging its masked language modeling approach to transfer knowledge across languages effectively. The major difference between BLOOM and XLM-R is that BLOOM emphasizes large-scale collaboration and multitasking, whereas XLM-R focuses on robust cross-lingual transfer and performance on multilingual benchmarks.

Feature	BLOOM	XLM-R	mT5	mBERT	mBART
Release Date	July 12, 2022	November 2019	February 2021	November 2018	January 2020
Number of Parameters	176 billion	Up to 10.7 billion	Up to 13 billion	110 million	406 million
Model Architecture	Decoder-only	Transformer	Encoder-Decoder	Transformer	Encoder-Decoder
Pretraining Data	ROOTS corpus (~11% code)	CC100 (167B tokens)	mC4 (6.4T tokens)	Wikipedia	Large text corpus
Multilingual Support	46 natural, 13 programming	100 languages	101 languages	104 languages	25 languages
Training Method	Pretraining, multitask finetuning (BLOOMZ)	Masked Language Modeling (MLM)	Span-corruption for text generation	Masked Language Modeling (MLM)	Denoising autoencoder objective
Zero-shot Learning	Strong performance	Effective cross-lingual transfer	Effective few-shot learning	Limited capabilities	Effective few-shot learning
Few-shot Learning	Significant improvement with few-shot examples	Robust performance	Robust performance	Moderate performance	Robust performance
Summarization Performance	High	Effective summarization	Effective summarization	Not specialized	Effective summarization
Translation Performance	Competent with right prompts	High performance	Good performance	Basic capabilities	Effective performance
Unique Features	Large-scale collaboration, multitasking, support for programming languages	Strong in low-resource languages, cross-lingual understanding	Robust across high-resource languages, versatile for text generation	General multilingual tasks, good zero-shot transfer	Multilingual translation, denoising autoencoder objective

Fig. 2: Comparative Analysis of Multilingual Language Models

3.3.5. PaLM (Pathways Language Model)

Google's PaLM is built to be extremely efficient at handling a wide range of linguistic jobs. It makes use of a sparse activation technique, which activates a small portion of the model's parameters at once. It uses the pathways system to activate specific parts of the model. This system allows a single AI model to handle multiple tasks efficiently. Instead of activating all parts of the model for every task, Pathways selectively activates only the parts that are most relevant to the current task. Because of this, PaLM is effective and scalable for a range of activities, including as understanding, translating, and creating text. The architecture of PaLM is optimized for few-shot learning scenarios and facilitates cross-lingual transfer.

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

3.3.6. GLaM (Generalist Language Model)

GLaM, is one of Google's other innovations. It is a mixture of experts. models(MoE) that dynamically selects subsets of the model's parameters for each input. Despite GLaM's large overall size of 1.9 billion parameters, only a subset (e.g., 145 million parameters) is used for each query, ensuring efficient computation. This architecture enables GLaM to achieve high performance with reduced computational cost. GLaM is particularly effective in handling diverse tasks across different languages, leveraging its modular design to adapt to specific language and task requirements. For each query, GLaM dynamically activates two of the most relevant experts tailored to the specific language and context. For instance, experts specialized in English language structures are activated for an English query, while different sets of experts handle Spanish and Mandarin queries.

The TABLE in Fig. 2 compares the key features of five language models: BLOOM, XLM-R, mT5, mBERT, and mBART. XLM-R is known for its strength in low-resource languages.mT5 performs well on a wide range of tasks, especially in high-resource languages.mBERT is a robust model for general multilingual tasks but has fewer parameters. mBART excels in translation and other sequence-to-sequence tasks.

3.3.7. Multilingual Models Designed for Indian Languages

Other models generally perform well in a broad range of languages but may not achieve the same level of accuracy for Indian languages. Whereas models that are optimized for Indian languages only outperform other general-purpose models like BERT, on Indian language tasks such as named entity recognition, part-of-speech tagging, and cross-lingual understanding [6].

MuRIL

MuRIL, which stands for Multilingual Representations for Indian Languages, is a language model developed specifically for Indian languages, covering 17 different Indian languages. It was created to tackle the problem of multilingualism faced by models in handling the diverse languages spoken in India. It can handle transliterated text (Indian languages written in the Latin alphabet), which is common in informal communication. MuRIL (Multilingual Representations for Indian Languages) is based on the Transformer architecture, similar to BERT, but with specific enhancements and training data tailored for Indian languages. MuRIL is trained on a large corpus of text specifically from Indian languages. This includes both monolingual text (text in one language) and bilingual text (translations between Indian languages and English).

AI4Bharat Models

AI4Bharat has developed a suite of models specifically designed to address the unique challenges of Indian languages. These models leverage the extensive datasets created by AI4Bharat and incorporate advanced techniques to handle the linguistic diversity of India.

- IndicBERT is a multilingual ALBERT model pre-trained in 12 major Indian languages. It uses a significantly smaller parameter count compared to models like mBERT, yet matches or exceeds their performance on various NLP tasks.
- IndicTrans2 is an advanced multilingual Neural Machine Translation model that supports all 22 scheduled Indian languages. It employs innovative techniques like script unification and transfer learning to improve translation quality, especially for low-resource languages.
- IndicNER is a Named Entity Recognition model fine-tuned for 11 Indian languages, capable of identifying entities across diverse domains.
- IndicCLS is a text classification model that can handle tasks like sentiment analysis and topic categorization across multiple Indian languages.
- The Indic-TTS project focuses on developing multi-speaker Text-to-Speech models for Indian languages, utilizing architectures like FastPitch and HiFi-GAN.
- IndicWav2Vec is an adaptation of Facebook's Wav2Vec model, pre-trained on large amounts of unlabeled Indian language audio data for speech recognition tasks.
- Lastly, IndicXLIT is a transformer-based transliteration model that can convert text between Roman script and native scripts for 21 Indic languages, trained on the extensive Aksharantar dataset. These models collectively form a comprehensive toolkit for processing and generating Indian language content, addressing various NLP tasks from basic language understanding to complex generation and translation tasks.

These AI4Bharat models form the foundation for building advanced multilingual voice assistants tailored to Indian languages. By leveraging these specialized models at different stages of the voice assistant pipeline, we can create a system that effectively handles the linguistic diversity of India[8]. Let's explore how these

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

models can be integrated into the key components of a multilingual voice assistant, addressing crucial tasks such as speech recognition, language identification, natural language understanding, machine translation, and speech synthesis. Each component plays a vital role in enabling seamless interaction across multiple Indian languages, and the AI4Bharat models provide the necessary tools to tackle the unique challenges posed by this linguistic landscape.

The system uses a shared vocabulary across languages and language-agnostic representations to process multiple languages efficiently. Models like IndicBERT and IndicTrans2 allow for seamless transfer learning and efficient language handling, enabling the voice assistant to adapt to different languages with minimal training. Additionally, specialized components like IndicLID and IndicXlit address the challenges of script diversity and Romanized text handling, which are common in Indian languages. This combination of models ensures the voice assistant can process, understand, and generate speech across India's linguistic landscape, making it inclusive of low-resource languages and varied scripts.

IV. COMPONENTS OF MULTILINGUAL VOICE ASSISTANTS

AI4 Bharat has made significant contributions to the development of multilingual models for Indian languages which has important applications in voice assistant technologies. Let's review some of their models involved in different stages of making a voice assistant.

4.1. Data Collection and Processing:

AI4Bharat has developed an extensive array of datasets and databases tailored for Indian languages, forming the backbone of their models. Their flagship IndicCorp is a monolingual corpus containing approximately 9 billion tokens from 12 major Indian languages, sourced from diverse online platforms. The Aksharantar dataset includes 26 million word pairs across 20 Indic languages for transliteration tasks. IndicNLG, another significant contribution, is a large-scale dataset for natural language generation tasks in 11 Indian languages, containing over 1.6 million samples for tasks like headline generation and text summarization. The IndicParaphrase dataset offers 38,000 sentence pairs across 10 Indian languages for paraphrasing and semantic similarity tasks. The AI4Bharat Speech Corpus provides a massive collection of transcribed audio data in multiple Indian languages, crucial for training ASR models [12]. Additional specialized datasets include IndicSentiment for sentiment analysis and IndicQA for question-answering tasks. The IndicNER dataset covers named entity recognition across 11 Indian languages.

The processing of this data involves several sophisticated steps. Initially, the collected text undergoes normalization to handle variations in spelling and script representation, which is particularly important for Indian languages. Unicode normalization is applied, followed by language-specific tokenization using tools like SentencePiece. BPE (Byte Pair Encoding) is employed for subword tokenization, especially effective for morphologically rich Indian languages. For speech data, spectral augmentation and SpecAugment techniques are applied to increase robustness. In machine translation tasks, pivot-based approaches generate parallel data for low-resource language pairs. Named entity recognition utilizes distant supervision techniques for automatic annotation of large text volumes [15].

AI4Bharat also employs active learning strategies to efficiently expand datasets where manual annotation is costly. The data undergoes rigorous quality checks, including language identification to filter mixed-language content and automated metrics to assess machine-generated data quality. For tasks like machine translation, parallel corpora are created by aligning sentences across languages. Innovative techniques like back-translation and transliteration are used to augment datasets, especially for low-resource languages. This meticulous approach to data collection and processing forms the foundation of AI4Bharat's state-of-the-art models for Indian language NLP tasks, enabling the development of powerful tools like IndicBERT, IndicTrans2, and IndicNER.

4.2. Automatic Speech Recognition:

ASR is a crucial step for any voice bot pipeline. It starts by processing the raw audio which is ineffective due to noise, so feature extraction techniques like Mel-Frequency Cepstral Coefficients (MFCCs) are employed [13]. MFCCs are key for extracting features from audio signals to improve machine learning tasks like speech recognition. The process involves converting the audio signal to digital, applying pre-emphasis to boost high-frequency energy, segmenting the signal into frames, applying the Discrete Fourier Transform (DFT), using a mel-filter bank to mimic human hearing, applying a log to the signal, and performing an inverse DFT. Dynamic features are also considered, resulting in 39 features per audio sample for the speech recognition model. MFCCs

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

transform audio into these sets of coefficients representing the power spectrum, which is better suited for machine learning. The next step is to process these coefficients using the ASR models.

AI4Bharat offers 2 main models at this stage. The first one is Indic Wav2vec which is an adaptation of Facebook's Wav2vec model. This model uses self-supervised learning for large amounts of unlabeled audio data, followed by fine-tuning on labeled data which makes it effective particularly for low-resource languages and scenarios where limited label data is available. Secondly, the Indic Conformer is a conformer-based speech recognition model that combines convolutional neural networks and transformers to achieve state-of-the-art performance in ASR tasks. It is trained in a supervised manner on approximately 30M parameters on datasets including ULCA, KathBath, Shrutilipi, and MUCS.

Wav2vec can achieve high accuracy when fine-tuned with labeled data, its accuracy is not similar to that of the Indic Conformer model considering that it is specifically optimized for a particular language. The Indic Conformer model has far better accuracy in Indic languages as it handles the linguistic complexities, such as phonetic diversity and context, present in these languages very well. Wave2vec models are faster during inference compared to Conformer models because they are designed to process raw audio in a more streamlined manner. Whereas, conformers tend to have more layers and complex computations during inference.

Once the MFCC features are extracted, the Indic Conformer model begins its work by using convolutional layers to capture local patterns in the audio signal. Next, it employs transformer layers to understand broader context and long-range dependencies in the speech. The model's output layer then generates probabilities for different characters or phonemes at each moment in the audio. These probabilities are decoded into text using methods like beam search or the Connectionist Temporal Classification (CTC) decoder. Finally, the text is refined through post-processing to correct errors and ensure it's formatted correctly, resulting in a polished and accurate transcription of the original audio input.

4.3. Language Identification:

For developing a voice assistant it is crucial to first identify the language that is given as input. Identifying languages can be tough, especially in low-resource languages. These languages don't have enough well-maintained data sets on which a model can be trained. AI4Bharat is helping by solving this issue by creating synthetic data for all the Indic Languages listed in the Indian Constitution both in the native script and Romanized text.

Recent advancements in language identification include Bhasha Abhijnaanam developed by AI4Bharat [10][14]. Bhasha Abhijnaanam is the state-of-the-art technique for creating training data sets for low-resource languages with the help of synthetic data spanning all 22 Indic languages which resulted in a new Language Identification Model (LID) called IndicLID. IndicLID is a language identification classifier that is designed for 22 Indian languages in both native-script and Romanized text. It is classified into 3 variants: a fast line classifier, a slower classifier fine-tuned from a pre-trained language model, and an ensemble of the two models that trade-off speed versus accuracy. The native-script model has better language coverage as compared to the existing LIDs.

On the other hand, the Romanized text model uses synthetic training data created via transliteration. IndicLID provides a significant contribution to the field of language identification for Indian languages, particularly in the context of romanized text, and serves as a basis for creating the NLP resources for Indian languages

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model developed by Google that processes text in both directions (left-to-right and right-to-left), allowing it to understand the context of each word by looking at surrounding words. BERT is pre-trained on large datasets and can be fine-tuned for specific natural language processing (NLP) tasks like text classification, sentiment analysis, and question answering. This makes it versatile for understanding and processing human language in different contexts.

ALBERT (A Lite BERT) is a more efficient version of BERT, designed to reduce the number of parameters while maintaining performance. It does this by reusing parameters across layers and using a factorized embedding method. This makes ALBERT faster and more memory-efficient, which is especially useful when dealing with multilingual models or large-scale applications.

IndicBERT is a multilingual adaptation of ALBERT, specifically fine-tuned for Indian languages. It is used in multilingual chatbots for several key reasons:

IndicBERT helps handle large corpora of text data from various Indian languages, enabling the collection

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

and organization of multilingual data. It can tokenize text across multiple scripts and convert it into a format ready for further processing, such as speech-to-text conversion or translation.

4.4. Language Understanding:

IndicBERT excels at understanding text in different Indian languages, making it ideal for tasks like intent recognition, sentiment analysis, and named entity recognition. It is designed to capture the linguistic diversity of India, which is essential for multilingual chatbots that need to understand and respond to users in different languages with accuracy.

By leveraging IndicBERT, multilingual chatbots can efficiently handle text processing and language understanding for a wide range of Indian languages, ensuring that chatbots can interact smoothly with users regardless of the language they speak.

4.5. Natural Language Understanding (NLU):

Their flagship model, IndicBERT is a multilingual ALBERT model pre-trained exclusively in 12 major Indian languages. IndicBERT is pre-trained on AI4Bharat's monolingual corpus, which consists of approximately 9 billion tokens, and is fine-tuned for specific tasks like intent classification and entity recognition. Despite having significantly fewer parameters compared to other multilingual models like mBERT and XLM-R, IndicBERT matches or exceeds their performance. This model's ability to understand and process multiple Indian languages has made an Indian-made multilingual voice assistant named Bhashni that can seamlessly switch between different Indian languages.

In the NLU stage, IndicBERT is employed to process and understand text across multiple Indian languages. IndicBERT, a multilingual ALBERT model, excels in tasks like intent recognition and entity classification, making it suitable for applications where users may switch between languages or use mixed-language inputs. Its shared vocabulary across languages and language-agnostic representations allow the model to capture the semantic meaning of text across different languages, ensuring efficient processing. IndicBERT's fine-tuning on Indian language datasets allows it to handle both native and Romanized scripts, providing robust language understanding for voice assistants.

4.6. Machine Translation

Translation is one of the key factors of multilingualism as it is the foundation for the whole working of the voice assistants to interact across different languages. In a country like India, users might prefer responses catered to their own languages. To make sure that their system can understand and respond to the user's preferred language, AI4Bharat's Indictrans2 is a robust multilingual transformer-based Neural Machine Translation model designed to handle translation tasks across all the 22 scheduled Indian languages including multiple scripts for low-resource languages. The model incorporates large-scale datasets and benchmarks, making it a comprehensive solution for multilingual machine translation in India.

It addresses the challenge of low-resource languages through script unification, where the model tries to unify the scripts that have similar linguistic roots like Hindi and Marathi, and transfer learning, in which the model uses knowledge from high-resource languages to improve the translations for languages with low resources by sharing its linguistic features. It leverages lexical sharing between languages with similar scripts, improving translation quality. Then the model will generate the translated text in the desired language by predicting the word sequence that resonates the most with the input's meaning.

4.7. Text Normalization and Transliteration

Text normalization and transliteration are crucial components for multilingual voice assistants, ensuring they can handle different scripts without losing meaning or pronunciation accuracy. Text normalization standardizes non-standard forms such as abbreviations, numbers, or dates into a uniform format. This is especially important for ensuring consistency when dealing with various languages and scripts. For example, it converts "Dr." to "Doctor" or formats dates uniformly. Transliteration, on the other hand, converts text from one script to another while preserving phonetic accuracy. This allows users to input or receive text in different scripts, ensuring that the meaning remains intact even if the script changes. For instance, a user may input text in Hindi (Devanagari script) and receive the output in Romanized text or another Indian script, without any loss in pronunciation.

AI4Bharat's IndicXlit model, based on the transformer architecture, is specifically designed to handle this task. It is trained on the Aksharantar dataset, the largest transliteration dataset for Indian languages, containing 26 million word pairs across 20 Indic languages [11]. IndicXlit uses a sequence-to-sequence model with attention mechanisms to map text between different scripts while maintaining pronunciation accuracy. This makes it highly effective for multilingual

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

applications that require seamless handling of diverse Indian languages and scripts. The model's multilingual functionality allows it to operate across a variety of Indic scripts, making it adaptable to different languages. Its ability to handle Romanized text and native scripts makes it a versatile tool for ensuring that voice assistants can work across diverse linguistic contexts while maintaining consistency and clarity in communication.

4.8. Response Generation (Text to Speech):

IndicBART is another model specially designed for sequence-to-sequence tasks in Indian languages [7]. It is based on mBART architecture and supports 11 major Indian languages. It can be used for various response generation tasks such as text generation, summarisation, question-answer generation, machine translation, etc. IndicBART is trained on a vast Indic language corpora of 452 million sentences and 9 billion tokens which also includes Indian English content. All the languages except English have been represented in the Devanagari script to encourage cross-lingual transfer learning between Indian languages [9].

The IndicBART model plays a crucial role in the response generation stage. IndicBART, a sequence-to-sequence model, is used to generate responses in various Indian languages. It is optimized for Indian languages, supporting multilingual output while maintaining fluency and accuracy across languages. This helps in generating responses that are contextually appropriate and fluent in the user's chosen language. The system leverages IndicBART's capability to manage complex tasks like summarization and question-answer generation, making the voice assistant more adaptable to the linguistic diversity of India.

Other than AI4Bharat models, Tacotron and Wavenet are often used in text-to-speech (TTS) systems to generate human-like speech.

4.8.1. Tacotron:

This is a neural network model that converts text into a sequence of speech features, later used to generate speech via a vocoder (like Wavenet).

4.8.2. Wavenet:

This model is a generative model of raw audio waveforms and is typically used as a vocoder to synthesize speech. It focuses on improving the quality of generated speech.

V. COMPARATIVE ANALYSIS OF MULTILINGUAL MODELS

Multilingual models have become crucial for voice assistants and language processing, yet several important areas still demand further exploration and improvement:

5.1. Performance on Low-Resource Languages:

While models such as mBERT, mBART, and XLM-R have made considerable progress, they still struggle with low-resource languages like Punjabi or Bengali. These languages often have insufficient labeled data, limiting the model's accuracy and performance. Future models need to focus more on developing techniques for low-resource scenarios such as zero-shot and few-shot learning to overcome this challenge. Data augmentation, synthetic data generation, and crowdsourcing are some potential methods to mitigate data scarcity.

5.2. Handling Code-Switching:

Many voice assistants encounter issues with code-switching, where users frequently switch between languages within a single conversation (e.g., a mix of English and Hindi). Multilingual models often find it hard to maintain context across languages. Existing models like mBERT and XLM-R could be fine-tuned to better handle this challenge. Researchers need to explore more robust ways to identify and manage these switches, such as creating specific datasets that feature natural code-switching in spoken and written forms.

5.3. Data Diversity and Representation:

While many multilingual models claim to support numerous languages, the quality of their performance often depends on the availability of training data. For instance, models like mT5 and BLOOM perform well in widely spoken languages but face issues in handling the linguistic nuances of underrepresented languages. Moreover, models trained in Indian languages, such as MuRIL and IndicBERT, have shown significant improvements in accuracy for tasks such as named entity recognition and part-of-speech tagging. However, expanding this approach to other regional languages could make multilingual voice assistants more inclusive.

5.4. Computation and Efficiency:

Larger multilingual models like PaLM (Pathways Language Model) and GLaM (Generalist Language Model) offer impressive capabilities but are computationally intensive, making them impractical for real-time applications such as voice assistants on low-powered devices. There is a need for more efficient models, perhaps through model compression

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

techniques or approaches like sparsity-based activations, which selectively activate parts of the model relevant to the current task. ALBERT (A Lite BERT) and IndicBERT offer good examples of smaller, efficient models that could be explored further.

5.5. Ethical Considerations in Multilingual NLP:

Multilingual models must address ethical concerns such as bias and fairness. Bias can be introduced through imbalanced datasets where high-resource languages dominate, leading to inaccurate or unfair outcomes in less-represented languages. This requires attention to fairness in model training and validation. Ethical AI development also involves privacy concerns, particularly with voice assistants, where sensitive personal data might be misused. Guidelines and regulations need to be defined for the ethical usage of multilingual voice assistants, ensuring they respect user consent and data privacy.

5.6. Emerging Research Trends:

The future of multilingual models should prioritize creating more comprehensive datasets, improving low-resource language support, and handling mixed-language inputs effectively. Models like IndicBART and IndicTrans2 could serve as a foundation for more inclusive models that handle the diverse linguistic landscape of countries like India while scaling up to more global applications. Additionally, addressing computational constraints through model optimization will be key to the deployment of these models in real-time applications.

VI. CHALLENGES IN MULTILINGUAL VOICE ASSISTANTS

Multilingual NLP consists of various challenges and creative solutions. The first major issue is linguistic diversity, as languages vary widely in vocabulary and grammar. Some are highly inflected, while others depend heavily on context. To tackle this, we've developed multilingual NLP models that can adapt efficiently to multiple languages. These models are trained to understand both the similarities and differences across languages. Researchers and software developers are finding ways to represent underrepresented languages by ensuring that NLP technology is inclusive.

Data scarcity is a significant obstacle for many languages, particularly ones with little digital content. This is especially difficult for languages with non-standard scripts or those spoken by smaller communities in rural areas. In order to overcome this, scientists are investigating few-shot and zero-shot learning strategies, which support models in operating effectively with minimal or no training data. Crowdsourcing and data augmentation are essential for increasing the amount of training data that is available for such languages.

The third problem is code-switching, where people switch languages within a conversation. For example "Aapko cars pasand hai?". This sentence is a combination of both Hindi and English. "Aapko pasand hai", translates to "do you like" in English. Thus, the whole sentence translates to, "Do you like cars?". Advanced models like multilingual BERT and GPT-3 have shown advancements in dealing with mixed-language text by leveraging their contextual understanding. There's also a big focus on fairness and bias, as NLP models can inadvertently reflect biases from their training data. Researchers are developing diverse datasets, bias-reduction techniques, and ethical guidelines to tackle this. Despite these challenges, ongoing research and innovation continue to improve the inclusivity, fairness, and capabilities of multilingual NLP [19].

VII. PROPOSED SYSTEM ARCHITECTURE

In this section, we propose a basic system for a multilingual voice assistant that uses the latest technology in speech recognition, language identification, machine translation, natural language processing, and speech synthesis. The goal is to build a system that can interact with users in different languages, regardless of their spoken language.

Currently, we are not using AI4 Bharat models as they are still in the development phase and they are not well documented. Instead, we are using other freely available open-source models as their alternative.

ISSN: 2456-2319
https://dx.doi.org/10.22161/eec.95.1

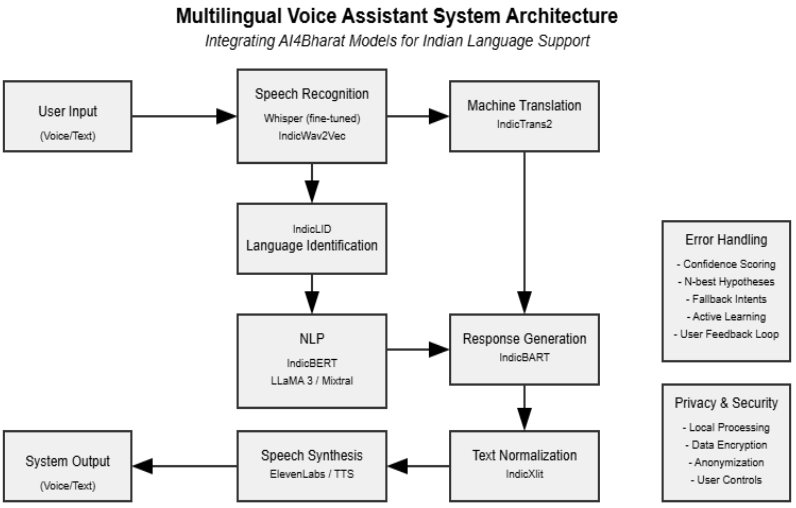

Fig. 3: Diagram of the proposed solution

7.1. Speech Recognition:

We use OpenAI's Whisper model to convert speech into text. Whisper is great at understanding different languages and accents, making it a good choice for multilingual applications. A version of Whisper that is fine-tuned for Indian languages performs better on datasets like Kathbath and CommonVoice. However, larger versions of Whisper offer more accuracy but require more computing power [20].

7.2. Language Identification:

A custom-built model is used to detect the language being spoken. Once the audio is received, it is converted into a mel-spectrogram, which helps identify the language by analyzing sound patterns. For text input, the *langdetect* tool can be used to determine the language.

7.3. Machine Translation:

Google's Translation API (googletrans) translates the input into English, which standardizes the language before it moves to the next step. While this allows for easier processing, it may sometimes lose the nuances of a language. An alternative solution would be to use models that understand multiple languages directly, without translating everything to English.

7.4. Natural Language Processing (NLP):

The translated English text is processed using the LLaMA 3 model. This model generates context-based responses. Other open-source models like Mixtral can be used for faster responses. If the translation step is skipped, IndicBERT can be used to handle non-English NLP tasks.

7.5. Response Generation:

After generating a response in English, we translate it back into the user's language using the *googletrans* API. This helps maintain the flow of conversation in the user's preferred language.

7.6. Speech Synthesis:

Finally, the translated text is turned into speech using ElevenLabs' voice synthesis tool. It can generate realistic-sounding speech in multiple languages [16]. If ElevenLabs isn't available, Google Text-to-Speech (TTS) can be used as a backup. Open-source tools like CoquiTTS, WavelAI, and MetaVoice can also be considered [17].

7.7. Data Storage:

We are storing input prompts and the generated response using MySQL. This works as a history to retrieve and show the previous chats in the front end.

The full implementation of the proposed system is publicly available on GitHub: https://github.com/soham1418/JARVIS

VIII. CHALLENGES IN MULTILINGUAL VOICE ASSISTANTS

In this paper, we have reviewed the main challenges of building multilingual voice assistants, especially for low-resource languages. We looked at the latest advancements in artificial intelligence, such as large language models and machine translation systems, and proposed an architecture that combines various new technologies to create a more inclusive voice assistant. Although current systems work well for popular languages, there's still a lot to be done to support underrepresented languages [18]. By focusing on

improving low-resource language support, better machine translation, and more efficient models, we can make voice assistants accessible to a wider range of people.

IX. FUTURE WORK

Going forward, several areas need more attention:

9.1. Improving Low-Resource Language Support:

More research is needed to develop better models for languages that don't have a lot of data. This includes methods like few-shot and zero-shot learning, which allow models to understand new languages with minimal data.

9.2. End-to-End Language Processing:

Instead of translating everything into English first, future systems should focus on understanding and processing multiple languages directly. This will help preserve the nuances of each language.

9.3. Efficient Model Designs:

Current models like Whisper and LLaMA are powerful but require a lot of computing power. Optimizing these models or developing lighter alternatives will make it easier to run voice assistants on everyday devices.

9.4. Adding Evaluation Metrics:

A proper evaluation of the proposed solution such as calculating BLEU Score, End-to-End Latency, Code-Switching Accuracy, etc.

REFERENCES

[1] Pavitra, A.R.R. et al. (2023) 'A Review on Intelligent Voice Assistant with Multilingual Support using Artificial Intelligence,' Journal of Emerging Technologies and Innovative Research (JETIR), 10(4). https://www.jetir.org/papers/JETIR2304137.pdf.

[2] Dong, Qianqian & Huang, Zhiying & Xu, Chen & Zhao, Yunlong & Wang, Kexin & Cheng, Xuxin & Ko, Tom & Tian, Qiao & Li, Tang & Yue, Fengpeng & Bai, Ye & Chen, Xi & Lu, Lu & Ma, Zejun & Wang, Yuping & Wang, Mingxuan & Wang, Yuxuan. (2023). PolyVoice: Language Models for Speech to Speech Translation. 10.48550/arXiv.2306.02982.

[3] ElevenLabs: Free Text to Speech & AI Voice Generator | ElevenLabs (2024). https://elevenlabs.io/.

[4] AI4Bharat (no date) GitHub - AI4Bharat/IndicWav2Vec: Pretraining, fine-tuning, and evaluation scripts for Indic-Wav2Vec2. https://github.com/AI4Bharat/IndicWav2Vec.

[5] Bark - a Hugging Face Space by Suno (no date). https://huggingface.co/spaces/suno/bark.

[6] Company, F. and Meta (2023) 'Introducing Voicebox: the most versatile AI for speech generation,' Meta, 16 June. https://about.fb.com/news/2023/06/introducing-voicebox-ai-for-speech-generation/.

[7] Dowding, John & Gawron, Jean & Appelt, Doug & Bear, John & Cherny, Lynn & Moore, Robert & Moran, Douglas. (1994). Gemini: A Natural Language System For Spoken-Language Understanding. 10.3115/981574.981582.

[8] Gupta, S.C. (2023) 'InDIC Language Stack for voice assistants and conversational AI | towards data science,' Medium, 6 August. https://towardsdatascience.com/vernacular-indic-language-bharat-bhasha-stack-for-conversational-ai-platform-and-voice-assistant-apps-6f8b9b4ad0a5.

[9] Dabre, R. et al. (2022) 'INDICBART: a pre-trained model for INDIC Natural Language Generation,' Findings of the Association for Computational Linguistics: ACL 2022 [Preprint]. https://doi.org/10.18653/v1/2022.findings-acl.145.

[10] Madhani, Y., Khapra, M.M. and Kunchukuttan, A. (2023) Bhasha-Abhijnaanam: Native-script and Romanized Language Identification for 22 Indic languages. https://arxiv.org/abs/2305.15814.

[11] Madhani, Y. et al. (2022) Aksharantar: Open Indic-language Transliteration datasets and models for the Next Billion Users. https://arxiv.org/abs/2205.03018.

[12] Bhogale, K.S. et al. (2023) VistaAr: Diverse Benchmarks and Training Sets for Indian Language ASR. https://arxiv.org/abs/2305.15386.

[13] Yadav, H. and Sitaram, S. (2022) A survey of multilingual models for Automatic Speech recognition. https://arxiv.org/abs/2202.12576.

[14] Rabiyath, S.S. et al. (2024) 'Bashini website and App – an overview,' Journal of Emerging Technologies and Innovative Research (JETIR), 11(1). https://www.jetir.org/papers/JETIR2401141.pdf.

[15] Mhaske, A. et al. (2022b) Naamapadam: a Large-Scale named entity annotated data for Indic languages. https://arxiv.org/abs/2212.10168.

[16] Create realistic Hindi Text to Speech | ElevenLabs (no date b). https://elevenlabs.io/languages/hindi.

[17] Auto Generate Hindi Voiceover Online | Wavel AI (2022). https://wavel.ai/solutions/ai-voice-generator/hindi-voiceover.

[18] Xiong, W. & Wu, L. & Alleva, Fil & Droppo, Jasha & Huang, Xuedong & Stolcke, A.. (2018). The Microsoft 2017 Conversational Speech Recognition System. 5934-5938. 10.1109/ICASSP.2018.8461870.

[19] Canbek, N.G. and Mutlu, M.E. (2016) 'On the track of Artificial Intelligence: Learning with Intelligent Personal Assistants,' Journal of Human Sciences, 13(1), p. 592. https://doi.org/10.14687/ijhs.v13i1.3549.

[20] Best Speech-to-Text APIs in 2024 (no date). https://www.edenai.co/post/best-speech-to-text-apis?referral=red-best-stt-apis-2Let.