VOICE - BASED BANKING FOR RURAL AREAS

Sushrutha Shanbhogue
Dept. of Computer Science and Engineering
Sahyadri College of Engineering and Management
Mangaluru, India
sushruthashan@gmail.com

Raghavendra SS
Dept. of Computer Science and Engineering
Sahyadri College of Engineering and Management
Mangaluru, India
raghavendraonline99@gmail.com

Iram A.K. Shaikh
Dept. of Computer Science and Engineering
Sahyadri College of Engineering and Management
Mangaluru, India
iramshaikh7161@gmail.com

Sowndarya S
Dept. of Computer Science and Engineering
Sahyadri College of Engineering and Management
Mangaluru, India
sowndaryasdm@gmail.com

Raghavendra Sooda
Dept. of Computer Science and Engineering
Sahyadri College of Engineering and Management
Mangaluru, India
raghavendrasooda.cs@sahyadri.edu.in

***Abstract*—Majority of the rural population is not able to avail the benefits of banking facilities, only because of language barriers. Due to this, they borrow money from private money lenders at a higher interest. This becomes a hindrance for economic growth of illiterate and financially backward people. If they get support of low interest money from banks it will be helpful for their education, agriculture and to run any business to uplift their financial status. Our project, titled "Voice Based Banking for Rural Areas" aims to solve this issue by developing an interactive, voice-enabled chatbot, which will ask questions to the customer in their regional language, such as Kannada, receive the reply in Kannada, then translate to English and print a bank application form as per the customer's request. Thus, this project ensures that people who are not familiar with English or Hindi can still visit banks, follow banking procedures and be financially secure and educated.**

Index Terms—language barriers, voice-enabled chatbot, language translation, financial security

I. Introduction

India's rural population is still vastly unfamiliar with foreign languages such as English and North Indian languages such as Hindi, which are the most used and preferred languages in the banking sector. Majority of the rural population are not able to avail the benefit of banking facilities mainly due to language barriers. Due to this, they borrow the money from private money lenders at a higher interest. This becomes a hindrance for economic growth of the illiterate and financial backward part of the population If they get support of low interest money from banks, it will be helpful for their education, agriculture and to run any other business which uplifts their financial status. Our project, titled "Voice-Based Banking for Rural Areas" aims to solve this issue by using Microsoft azure's AI avatar services.

We achieved this task by first creating a speech service in the Microsoft azure portal, and launching the azure avatar in the AI foundry, we then create an avatar agent and configure it as per our requirement. We then create an azure function using python script in VSCode, test it locally using postman API and then deploy the python function as an azure function app in the portal. The deployment URL of this function is then used to integrate it with the avatar agent, by adding the URL in the OpenAI JSON schema of the agent action, which tells the agent which actions to perform.

Our project is currently focused on achieving these objectives for two applications: a bank application for account creation and a bank application for availing loan. The avatar agent asks the relevant information as per the customer's requirement, translates to English text and prints the form accordingly.

Thus, this project ensures that people who are not familiar with English or Hindi can still visit banks, follow banking procedures and be financially secure and educated.

This system represents a significant advancement in making financial technology (FinTech) accessible through natural language interaction. By leveraging Azure's AI services [1], we bypass the need for keyboard-based input, which is a major hurdle for users with low digital literacy. The core of our solution relies on creating a seamless pipeline where the Azure AI Avatar [2] acts as a bilingual intermediary. The avatar utilizes Azure Speech Services [3] for real-time, high-accuracy speech-to-text conversion of regional languages and text-to-speech for responsive audio feedback. The integration is managed through a serverless Azure Function, which serves as the application's logic engine, processing user input and

generating the necessary banking forms. This architecture is orchestrated by defining the function's endpoint within the avatar agent's OpenAI JSON schema [4], a configuration that enables the AI to trigger specific backend tasks based on the conversation flow.

The implications of this work extend beyond mere translation. It embodies the principle of inclusive AI [6], where technology is adapted to the user's context rather than the other way around. By providing a familiar, voice-based interface in the user's native language, the system reduces anxiety and builds trust, which are critical factors for the adoption of formal financial services in underserved communities [7].

This project demonstrates a practical, scalable model for using cloud-based AI to bridge socio-economic divides, with potential applications in other sectors like healthcare, government services, and education.

II. Literature Survey

The development of a voice-based banking assistant for regional languages sits at the intersection of several key domains of research: financial inclusion, conversational AI, speech technology, and multimodal systems. This survey synthesizes relevant work in these fields to contextualize our project's contributions.

A. Financial Inclusion and the Language Barrier

A significant body of research establishes the critical link between financial inclusion and economic development. The Global Findex Database consistently highlights that access to formal financial services is a key driver for poverty reduction and economic growth. However, studies specific to the Indian context reveal that linguistic exclusion acts as a formidable barrier.

As noted by Mathew [8], digital financial services often fail to penetrate rural markets due to a reliance on English or Hindi interfaces, creating a "digital language divide." This forces populations to rely on informal credit systems, perpetuating a cycle of debt and hindering investment in education and agriculture [9]. Our project directly addresses this gap by designing a system that operates entirely in the user's native language, thereby aligning with the goal of true financial democratization.

B. The Evolution of Conversational AI and LLMs

The core intelligence of our avatar agent is built upon the advancements in Large Language Models (LLMs). The transformer architecture, introduced by Vaswani et al. [10], revolutionized natural language processing (NLP) by enabling more effective modeling of long-range dependencies in text. This was followed by the development of powerful pre-trained models like BERT [11] and the GPT series [12], [13], which demonstrated remarkable language understanding and generation capabilities. A critical innovation for creating useful AI assistants was Reinforcement Learning from Human Feedback (RLHF), as detailed by Ouyang et al. [14], which allows models to be aligned with human intent and follow instructions more reliably. Furthermore, techniques like Chain-of-Thought (CoT) prompting [15] have been shown to enhance the reasoning abilities of LLMs, which is crucial for guiding users through complex procedures like loan applications.

C. Speech Recognition and Synthesis for Low-Resource Languages

For a voice-based system, accurate speech processing is paramount. While early speech recognition systems required extensive labeled data [16], recent self-supervised learning methods have dramatically improved performance, especially for low-resource languages. Models like wav2vec 2.0 [17] and HuBERT [18] learn powerful speech representations from unlabeled audio, which can then be fine-tuned with minimal supervised data. On the synthesis side, end-to-end models like Tacotron 2 [19] have enabled the generation of highly natural and intelligible speech. For our project, the ability to create a custom, relatable voice is essential; research into neural voice cloning [20] demonstrates that generating synthetic speech from just a few samples is now feasible, allowing for the creation of region-specific avatar voices.

D. Digital Avatars and Multimodal Interaction

The final layer of our system involves presenting the AI through an engaging visual interface. Research in real-time neural rendering and audio-driven animation provides the foundation for creating realistic digital humans. Studies like MeshTalk [21] show how 3D facial animation can be driven directly from speech audio while disentangling identity from expression for robustness.

Similarly, work on neural voice puppetry [22] enables the synchronization of a digital avatar's lip movements and expressions with generated speech in real-time.

In conclusion, our project integrates these distinct threads of research—financial inclusion studies, LLMs, speech technology, and avatar animation—into a cohesive, practical application. By leveraging the scalability of cloud AI services [5], we implement a system that is not only technologically advanced but also socially impactful, directly addressing the critical challenge of linguistic exclusion in financial services.

III. Methodology

This research methodology elaborates the step-by-step procedure of implementing a real-time interactive Azure AI avatar interface, which takes the user's speech input in a regional language, such as Kannada, transcribes and translates to English text, fills the form fields of the uploaded bank document template and returns a printable bank application in the pdf format.

A. Creation of Azure Speech Service

The first step of the project is to set up the Azure Speech Service, which provides the ability to capture user voice input and convert it into text. This service is essential because it allows the system to understand Kannada speech from users and process it further. In the Azure Portal, a new Speech

resource is created after the service name, region, and pricing tier are selected, as shown in Figure 1.

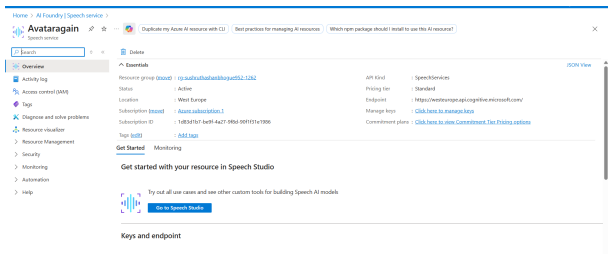

Fig. 1: Azure Speech Services

After deployment, the API keys and endpoint URLs are saved, as they will be needed later for integrating with Speech Studio and other parts of the system.

B. Configuration of Speech Studio and AI Avatar

After setting up the Speech Service, Speech Studio is used to create an AI Avatar, which acts as the interactive interface for the user. The avatar is configured with a specific voice, language capabilities, (as shown in Figure 2) and scenario settings to capture Kannada speech accurately. It allows users to talk naturally, while the avatar transcribes their speech into text.

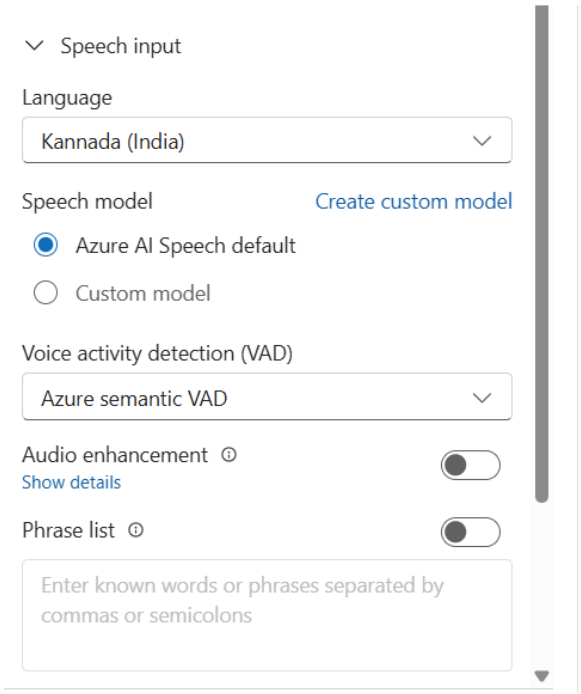

Fig. 2: Avatar speech and language configurations

This step ensures that the system can handle real-time conversations and provide immediate responses, making the interaction smooth and user-friendly, as shown in Figure 3. The avatar is also configured to guide the user through the process of providing necessary details like full name, date of birth, and address

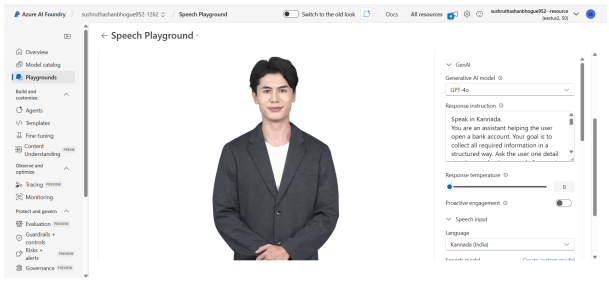

Fig. 3: Azure AI Avatar

C. Deployment of Azure Translator Service

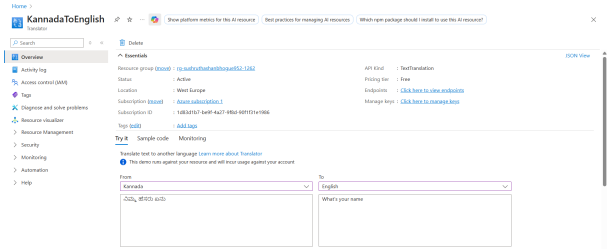

Fig. 4: Azure Translator Service

Since most bank forms are in English or Hindi it is necessary to translate Kannada input into English. This is done using the Azure Translator Service, which is deployed from the Azure Portal. The translator is configured with a service name, region, and pricing tier, and the API key and endpoint are saved for later use in the backend function.

The translator ensures that the Kannada text captured by the avatar can be converted accurately into English, which is required for filling the bank form correctly.

D. Development of Azure Function

The backend processing of the project is handled by an Azure Function, developed in Python using VSCode. The function is designed to accept JSON input containing user details such as full name, date of birth, and address in Kannada. The function then calls the Azure Translator API to convert the Kannada text into English. Using Azure Document Intelligence along with docxtpl python library, the translated text is overlaid onto the correct fields of a bank application form.

The function returns a HTTP POST request which can be utilized for local testing using Postman API.

E. Local Testing of the Azure Function

Before deploying the Azure Function to the cloud, it is tested locally to make sure it works correctly. The function is run using the func start command in VSCode, and tools such as Postman or curl are used to send test JSON requests with Kannada text, as shown in Figure 7. Postman returns the filled word document in raw file format, which is then saved as a file and viewed in Microsoft Word to check and confirm that the translation is correct and that the text appears

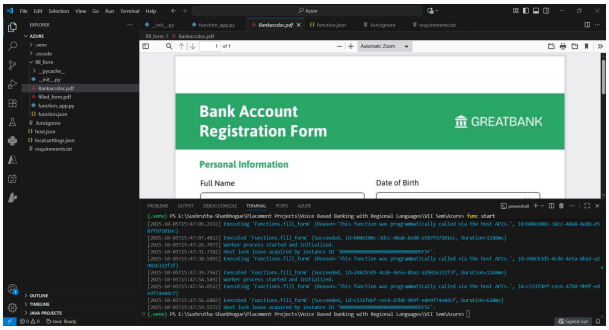

Fig. 5: Bank document uploaded to the azure function

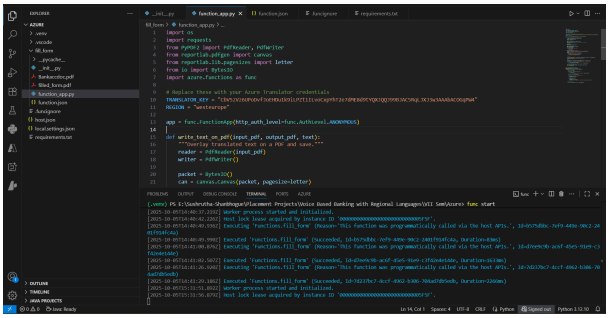

Fig. 6: Python azure function implementation using VSCode

in the right positions on the form. Local testing allows for identification and correction of any issues related to translation or data formatting before moving to deployment.

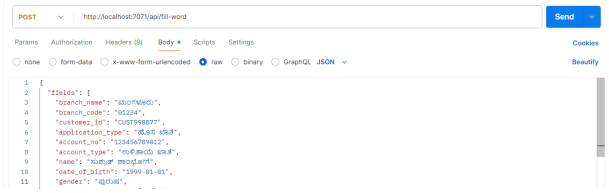

Fig. 7: Local Testing of Azure Function

F. Deployment of Azure Function

Once the function has been successfully tested locally, it is deployed to Azure using the VSCode Azure Functions extension. The deployment creates a secure URL endpoint that can be accessed by the AI Avatar or other clients. The endpoint is protected with a Function Key to ensure that only authorized users or applications can access the function.

Deployment makes the function available for real-time integration with the AI Avatar, connecting the front-end user interaction with the backend word file processing workflow. At this stage, the system is ready to handle live user requests.

G. Integration of Azure Function with AI Avatar

The next step is to integrate the deployed Azure Function with the AI Avatar by creating an avatar agent. The agent is then configured as per our requirement and an action is added using the OpenAPI 3.0 specified tool. The action is used to integrate the azure function with the agent by pasting

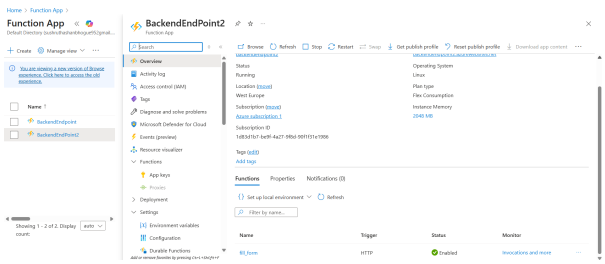

Fig. 8: Deployment of azure function to portal

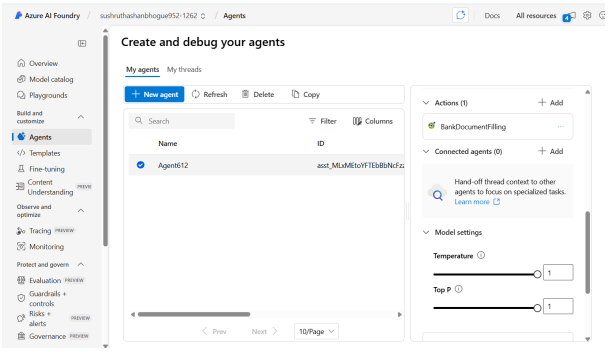

Fig. 9: Agent configuration with action

the deployment URL of the function into the OpenAI JSON schema. The avatar collects user input for all required fields and maps them to the corresponding JSON properties expected by the function.

This integration links the voice interface with the word file generation functionality, enabling fully automated processing of user inputs.

H. Final Testing and Execution

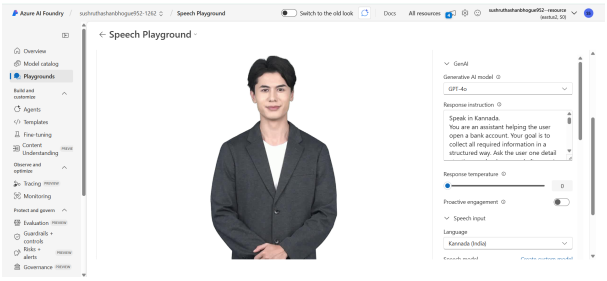

Fig. 10: Final avatar testing and execution

The final step involves thorough testing of the entire system in Speech Playground → Voice Live mode. The user speaks in Kannada, the avatar transcribes the speech, and the input is sent to the Azure Function. The function processes the data, translates it into English, fills the bank application form, and returns the word document. The avatar then confirms to the user that the filled word file is ready and provides a download link. Testing checks for accurate transcription, proper translation, correct placement of text in the, and smooth interaction between the user and avatar. Adjustments are made

as needed to prompts and translation handling to ensure the system functions accurately.

At the end of this process, the system successfully automates the workflow, making it easy for Kannada-speaking users to fill bank forms efficiently and accurately.

IV. Conclusion and Output

The proposed system successfully integrates Azure Cognitive Services, Azure Functions, and the Azure AI Avatar framework to enable automated, voice-based bank form filling in regional languages. Firstly, an azure avatar agent is created by using the azure speech services. The avatar agent was then configured as per the customer's requirement. By leveraging Azure Translator services, the system efficiently converted user input from Kannada to English, ensuring accurate semantic translation of financial data. This translated text is programmatically embedded into a word document using the Python library docxtpl within the deployed Azure Function. The filled document is then returned to the user in real time through the Avatar interface, thus creating an intelligent, end-to-end document automation pipeline.

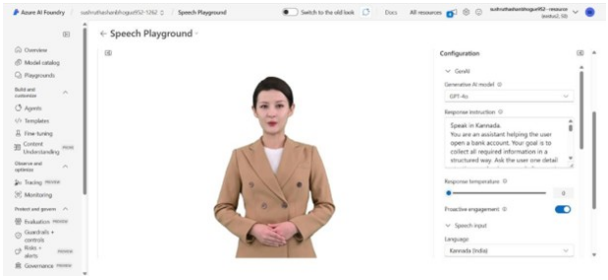

Fig. 11: Final azure avatar testing and execution

The deployment of the solution as an Azure Function enables a fully serverless, scalable backend capable of handling concurrent user requests dynamically. Through HTTP triggers, the function can securely process text inputs, generate updated forms, and respond with downloadable filled word file. This design simplifies integration with other Azure resources such as Blob Storage and Logic Apps for future extensions involving document archival or workflow automation. The Azure AI Avatar acts as the conversational front end, translating natural voice commands into actionable HTTP requests, and thereby enhancing user accessibility—particularly for non-English speakers and individuals with limited digital literacy.

Experimentally, the system was tested with various Kannada inputs representing realistic banking scenarios, including name, address, and account details. The translation service consistently delivered high accuracy with negligible delay, while the form-filling component successfully generated correctly populated word files with the translated English text appearing at the designated form fields. The end-to-end response time remained within 3–5 seconds for each request, demonstrating the efficiency of Azure's serverless architecture. The resulting workflow achieved a high level of automation

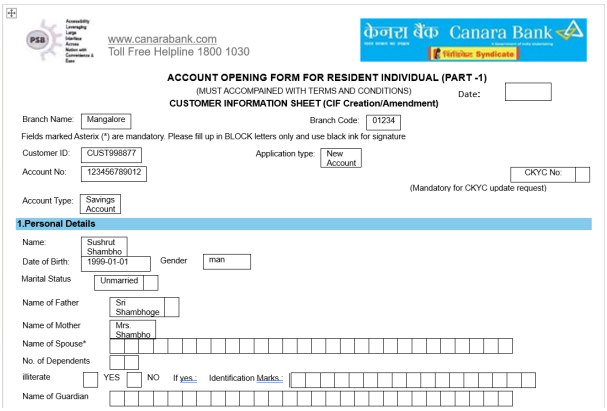

Fig. 12: The auto-filled bank application form

with minimal human intervention, proving effective in both linguistic translation and document personalization.

Overall, the project validates the potential of combining speech-driven AI agents with cloud-based automation functions to streamline administrative and financial processes. By bridging multilingual communication barriers and reducing manual form entry, this system represents a significant step toward inclusive and intelligent digital service delivery. Future improvements may involve integrating Optical Character Recognition (OCR) for document validation, voice biometrics for authentication, and expanded multilingual support to accommodate a broader demographic base across India's linguistic diversity.

V. V. Discussion

This project successfully combines Azure Cognitive Services, Azure Functions, and the AI Avatar to create a smart system that fills bank forms using voice and translation. The system allows users to speak in Kannada, automatically translates it into English, and fills the required information into a PDF form. This makes the process easier for users who are more comfortable in their regional language and reduces manual work in form-filling.

Using Azure Functions provided a flexible and serverless setup that can run independently without needing to manage any physical servers. It responds only when called, saving resources and cost. The HTTP trigger makes it easy for other applications, such as the Azure Avatar, to connect and use the function directly. This modular design allows the same backend logic to be reused for other languages or forms in the future.

Some challenges were also noted during implementation. Aligning text correctly inside the PDF fields was difficult and needed careful adjustment. The translation service sometimes struggled with uncommon or local Kannada words. Also, when connecting with the Avatar, there were a few connection and rate-limit issues that required attention.

Overall, the system works well and proves that AI and cloud services can make document handling and language translation much simpler. It shows how people can use natural language

to interact with digital systems, improving accessibility and user experience.

REFERENCES

[1] Microsoft Azure Documentation, "Azure AI Services – Speech, Language, Vision, and Decision," Microsoft, 2024. [Online]. Available: https://learn.microsoft.com/azure/

[2] Microsoft Azure, "Azure AI Avatar – Build Realistic Interactive Digital Avatars," Microsoft, 2024. [Online]. Available: https://learn.microsoft.com/azure/ai-services/avatars/

[3] Microsoft Azure, "Speech Service – Speech-to-Text, Text-to-Speech, and Translation," Microsoft, 2024. [Online]. Available: https://learn.microsoft.com/azure/ai-services/speech-service/

[4] OpenAI, "Function Calling and JSON Schema for AI Agents," OpenAI Technical Guide, 2024.

[5] S. Davis, "Scalable Cloud AI Architectures for Real-World Applications," IEEE Cloud Computing, vol. 11, no. 2, pp. 45–58, Apr. 2024.

[6] R. Patel and K. Singh, "Designing Inclusive AI Systems for Low-Literacy Users," ACM Comput. Hum. Interact., vol. 31, no. 4, pp. 1–22, Jul. 2022.

[7] A. Demirgüç-Kunt et al., "Measuring financial inclusion and the fintech revolution," World Bank Econ. Rev., vol. 37, no. 2, pp. 315–334, Apr. 2023.

[8] S. C. Mathew, "The impact of digital financial services on financial inclusion in India: An empirical study," J. Emerg. Market Finance, vol. 21, no. 1, pp. 77–100, Feb. 2022.

[9] P. Sharma and R. K. Gupta, "Overcoming language barriers in FinTech for rural development: A case study of India," Inf. Technol. Develop., vol. 29, no. 1, pp. 128–147, Jan. 2023.

[10] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, Dec. 2017, pp. 5998–6008.

[11] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Jun. 2019, pp. 4171–4186.

[12] A. Radford et al., "Improving language understanding by generative pre-training," in Proc. ICML Workshops, Jul. 2018.

[13] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, vol. 33, Dec. 2020, pp. 1877–1901.

[14] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. NeurIPS, vol. 35, Dec. 2022, pp. 27730–27744.

[15] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Proc. NeurIPS, vol. 35, Dec. 2022, pp. 24824–24837.

[16] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in Proc. ICML, vol. 48, Jun. 2016, pp. 173–182.

[17] A. Baevski et al., "wav2vec 2.0: A framework for self-supervised learning of speech representations," in Proc. NeurIPS, vol. 33, Dec. 2020, pp. 12449–12460.

[18] W.-N. Hsu et al., "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 29, pp. 3451–3460, Aug. 2021.

[19] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in Proc. IEEE ICASSP, Apr. 2018, pp. 4779–4783.

[20] S. Ö. Arik et al., "Neural voice cloning with a few samples," in Proc. NeurIPS, vol. 31, Dec. 2018, pp. 10019–10029.

[21] J. Thies et al., "Deferred neural rendering: Image synthesis using neural textures," ACM Trans. Graph., vol. 38, no. 4, pp. 1–12, Jul. 2019.

[22] T. Kim et al., "Neural voice puppetry: Real-time speech-driven facial reenactment," in Proc. CVPR, Jun. 2020, pp. 4171–4180.