ALDI Talk

Adaptive Language/Dialect Interpreter for Elderly

Module group - T2

Team Name - ALDI Talk

Date - 9 Feb 2025

Name	Admin number
Min Phyo Thura	233523A
Lim Jin Bin	221128Z
Alexander Chan Rui Bin	230648A
Mohammad Habib	231880L

Table of Contents

Introduction

Company Background: Apex Harmony Lodge (AHL)

Apex Harmony Lodge (AHL), established in 1999, is Singapore's first purpose-built home dedicated to individuals with dementia. AHL's mission is to empower the lives of individuals affected by dementia through a strength-based, person-centred approach to integrated care. The facility provides both residential and community care services, reframing dementia as a unique and celebratory phase of life.

Services Provided

AHL offers a variety of tailored care services, which include:

- **Residential Care**:
  - *Assisted Living (AL)*: Designed for residents who are cognitively and physically able, promoting a lifestyle centred on ability and engagement.
  - *Supported Living (SL)*: For residents requiring some assistance with daily activities, supported by community interaction and assistive devices.
  - *Tender Loving Care (TLC)*: Providing comfort and unconditional support for residents in the later stages of dementia.
- **Community Care**: Programs such as Therapy Through Work (TTW), EmbrACE, and Caregiver Mastery aim to integrate persons with dementia into the wider community while offering support to caregivers.

Problem Statement

Despite the broad spectrum of services offered, AHL faces significant communication gaps between staff and residents. This issue arises because caregivers often communicate primarily in standard English, while many elderly residents express their needs in local dialects, such as Hokkien or Cantonese. As a result, residents may struggle to articulate their concerns clearly, leading to delays, misunderstandings, and potential lapses in care quality.

Business Objective

The objective of this project is to develop a direct speech-to-speech translation tool that improves communication between residents and caregivers at AHL. This tool will support Singapore-focused languages such as Mandarin, Malay, Tamil, and local dialects like Hokkien and Cantonese. By minimizing communication delays, the solution aims to enhance the overall quality of care, align with AHL's person-centred mission, and improve the well-being of both residents and staff.

Solution Approach

AldiTalk uses a three-stage approach to address direct speech-to-speech translation: transcription of the user's speech (STT), machine translation, and generating a translated audio (TTS).

Phase 1: Language Selection

Singapore's elderly population is often multilingual, with many capable of speaking at least two languages. However, dialects remain a critical factor in effective communication for elderly care. According to CNA, around half of those who speak Chinese dialects use Hokkien while another quarter use Cantonese. AldiTalk, prioritising communication support in elderly care, proposes a scalable approach to implementing new custom translation models and incorporating them into existing language models across the following commonly spoken languages and dialects:

- **Standard Languages**: English, Mandarin, Malay, Tamil, Hindi
- **Dialects**: Hokkien, Cantonese

The inclusion of both widely-used languages and underrepresented dialects ensures the tool can address the diverse linguistic needs within care facilities like AHL.

Phase 2: Existing Solutions & Challenges

Azure AI services mostly cover commonly used standard languages including English, Mandarin, Malay, Tamil, Hindi, and even Cantonese, the second most-used Chinese dialect in Singapore. While these solutions can easily be incorporated into AldiTalk, a big challenge remains for those unsupported but highly in-demand dialects like Hokkien.

Phase 3: Data Collection & Development of Hokkien Model

Hokkien is a popular dialect in much use, especially in Singapore and Taiwan. However, its nature of being a fully oral language (i.e., no standardised written format) makes it challenging to develop a translation model. Our approach is to create a Hokkien STT model that takes in Hokkien audio and outputs Mandarin transcripts, given their nature of little but tiny similarity. We will convert publicly available Hokkien data and personally recorded audio into a format suitable for Azure (Mandarin) Custom Speech, which will be fine-tuned with lesser weights from the base Mandarin model(30%) but more from the Custom Hokkien caregiving model(70%). For Hokkien TTS, since Azure's Custom Neural Voice is expensive and not easily accessible, we will use Meta's existing mms-tts-nan model, which is a TTS model for the Chinese Min Nan dialect. It is not fully Hokkien but is closest and most suitable for Hokkien at this point in time.

Phase 4: Cross-platform compatibility & UI/UX

We had previously proposed that we intend to use Flutter, which is the optimal choice to develop a cross-platform native application. But, Flutter is not a beginner-friendly language, and thus we opted to make AldiTalk a web application that is accessible on any device, any platform or any browser, using HTML, CSS, JS and Python. UI is a minimalist theme, which is not distracting for old seniors, and for user-friendliness, we believe that taking as few clicks as possible to get the translation done is the most optimal UX for elderly use. AldiTalk will be designed to translate any language, whether speech or text, in just a single click.

Data Collection & Analysis

Among the languages listed above, most are already supported by Microsoft's Azure Speech and Translation services. This allowed us to concentrate our development efforts primarily on Hokkien, a low-resource dialect not natively supported by Azure. Developing a reliable Hokkien speech translation model required acquiring and enhancing a quality dataset of sufficient size. However, data availability for Hokkien was a significant challenge. To overcome this, we employed two primary data collection methods:

1. Hokkien dramas from YouTube (MediaCorp and Gov.sg)

  - **Video Segmentation**: We segmented each video into individual frames to identify scenes with clear, defined transcripts for accurate subtitle extraction. Frames with sharp edges were prioritized to improve OCR accuracy.
  - **Optical Character Recognition (OCR)**: Using the Azure AI Vision service, we performed OCR on the selected frames to extract both English and Mandarin subtitles embedded within the video. This process allowed us to generate transcript data for each frame.
  - **Transcript Alignment**: The extracted English and Mandarin subtitles were matched with their corresponding video frames to synchronize text with spoken audio accurately.
  - **Audio Segmentation**: Once the transcripts were aligned with the video frames, we split the audio into smaller, time-matched segments. These segments were then used to train the Hokkien speech-to-text model with Azure Custom Speech (Mandarin).

2. Speech data from volunteers (friends, family, lecturers)
   - **App Development**: We created an app specifically designed for recording Hokkien speech. The app provided a simple interface for volunteers to participate in the data collection process remotely.
   - **Mandarin Prompt Instructions**: Once the app was set up, volunteers were given a Mandarin phrase on-screen. Their task was to read out the Hokkien equivalent of the phrase, ensuring authentic language usage in the recordings. This method helped capture dialect-specific vocabulary and phrasing that might not be present in standard datasets.
   - **Data Submission**: After completing their recordings, volunteers submitted the audio files through the app. These recordings were then sent back to us for integration into the training dataset.
   - **Data Augmentation:** applied various data augmentation techniques to increase its size and variability. These techniques included:
     - Adjusting **audio speed** to simulate different speaking rates.
     - Modifying **pitch** to account for variations in voice tone and intonation.
     - Altering **volume levels** to reflect different vocal intensities.

    - Adding **background noise** to mimic real-world environments such as care facilities and coffee shops.

After data collection, we gathered a total of **21 hours** of speech data from YouTube videos featuring Hokkien conversations and **8 minutes** of high-quality speech data recorded by volunteers. However, the initial volunteer dataset was insufficient for effective training of the speech recognition model. Therefore augmentation was applied to expand the dataset to **3 hrs.**

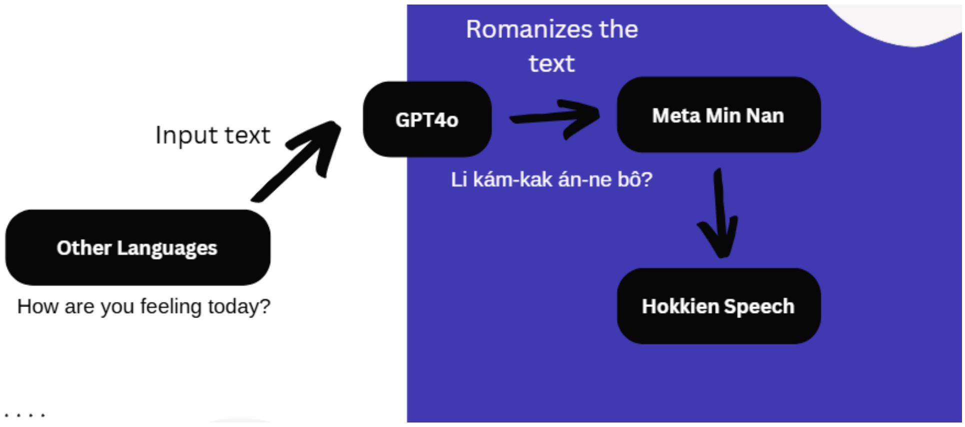
AI Solution Development

Behind-the-scene AI models for AldiTalk greatly leverage Azure AI services and custom integrations to facilitate real-time speech-to-speech (S2S) translation. The system uses Azure's Speech, Translation, and OpenAI services to handle supported languages, while additional services and models were implemented to support Hokkien.

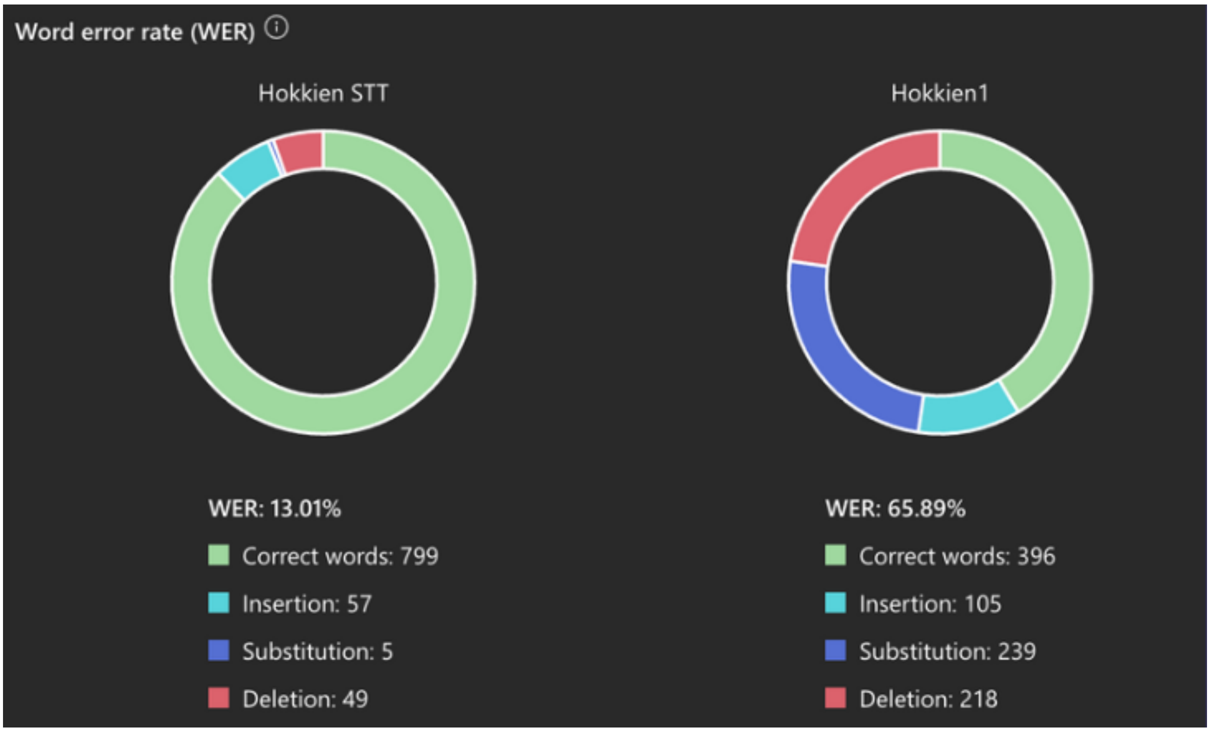
Core Azure AI Components

- **Azure Speech-to-Text (STT)**
  - Converts spoken input from users into text in real-time.
  - Once the user speaks, the API processes the input and returns recognized text after analyzing the audio.
  - To improve usability, silence timeout settings are adjusted to accommodate longer speech inputs, reducing the risk of interrupted transcriptions.
- **Azure Translation Service**
  - Once text is generated from STT, it is sent to the translation API to be converted to the target language
- **Azure Text-to-Speech (TTS)**
  - The translated text is sent to the TTS API to generate spoken output in real time.
  - Neural voices, such as "Jenny" for English or "Xiaoxiao" for Mandarin, are used to ensure natural and clear audio synthesis.

- **Azure Custom Speech (Hokkien STT)**
    - Convert spoken Hokkien from both pre-recorded and live recordings into Chinese representations of Hokkien
    - Chinese encoded Hokkien carries identical meaning to the actual spoken phrases
- **Custom Azure OpenAI Integration**
    - used to handle specialized text generation, particularly for unsupported dialects like Hokkien.
    - Input sentences are converted into Pe̍h-ōe-jī (POJ), a Romanized representation of Hokkien, to maintain phonetic accuracy.
    - System prompts are also provided to instruct the AI (GPT4o) to act as a Hokkien Language expert, ensuring the correct tone and character usage.
- **Meta MMS-TTS-NAN (Hokkien TTS)**
    - For Hokkien speech output, the solution integrates a custom TTS model from Meta through Hugging Face API.
    - The Romanized text produced by GPT-4o is processed by the Meta TTS model to generate Hokkien (Chinese Min Nan) audio.
    - This allows the solution to provide natural-sounding Hokkien speech output, compensating for Azure's lack of native Hokkien TTS support.



Model Evaluation & Interpretation



Our first model, **Hokkien1**, was trained solely on the collected YouTube data. This model exhibited a **high Word Error Rate (WER) of 65.89%**, indicating poor recognition accuracy. This result highlighted the limitations of relying solely on publicly sourced data, which may contain background noise, overlapping speech, and inconsistent audio quality.

This led us to use our second data collection method of gathering targeted speech samples from volunteers. These recordings provided clear and dialect-specific data, addressing the deficiencies in the original dataset.

Our second model, **Hokkien STT**, was trained on a combination of both YouTube and volunteer-collected data. The inclusion of augmented, high-quality speech samples led to a **significant reduction in WER to 13.01%**, indicating major improvements in recognition accuracy.

System Deployment & Maintenance

Our application is built using Flask and follows a modular, secure, and scalable architecture. Key aspects of the deployment and integration are as follows:

Flask Deployment:

The application is developed with Flask, which serves HTML templates and static assets from designated folders.

Flask creates a WSGI-compliant application object (via app = Flask(__name__)), enabling both development and production deployments.

Backend Integration:
RESTful endpoints are defined in app.py that process client requests and return JSON responses.

These endpoints invoke functions from AldiTalk/alditalk.py to interact with external services (e.g., Azure Cognitive Services) for translation and speech functionalities.
Configuration and Security:

Sensitive configuration details such as API keys and endpoints are stored in a .env file.

The application loads these credentials using the python-dotenv library, ensuring secure and flexible management without hard-coding sensitive information.
Production-Ready Deployment:

During development, the Flask app is run using the built-in server (via app.run(debug=True)), which is suitable for testing and debugging.
For production, the inherent WSGI compliance allows deployment via production-ready WSGI servers (e.g., Gunicorn or uWSGI) behind a reverse proxy like Nginx, providing enhanced performance, scalability, and security.

This approach ensures that our application remains robust, secure, and scalable, with a clear separation between the front-end and back-end components while leveraging industry-standard tools for deployment.

Discussion & Conclusion

Challenges and Limitations

1. Lack of native Hokkien support in Azure services.
2. Limited availability of data for Hokkien speech and text.
3. Budget and subscription limitations that affected data processing and model training. → (OCR on average takes $2 per episode and model training is $10/hr)
4. Insufficient outreach and research resources to support underrepresented dialects.

Potential Improvements

1. Supporting additional dialects and languages.
2. Improving the Hokkien model by collecting more diverse data from targeted sources like the Hokkien Foundation
3. Enhancing context-awareness to better handle caregiving scenarios.
4. Compatibility with wearable technologies such as smart earphones and smart glasses.

Objectives Met

1. Our app supports and translates the major languages/dialects spoken by Singaporean seniors.
2. Our app is very easy to use:
    - huge and minimal amount of buttons.
    - Dark/light mode for comfortable viewing
3. We have successfully developed a portable system accessible via smartphones and tablets to improve caregivers' responsiveness.

Appendices

AldiTalk: Connecting Voices, even the Silent Ones

Our Data

Ready-to-train Hokkien audio segments with Mandarin transcript

Raw frames

Raw .wav audio

Raw OCR results

Cleaned OCR results

Codes

References

CNN News Article

YouTube Playlists

Hokkien Test

Subtitles Test

WhatEverWillBe

Towkay

EatAlready

HoSehBo

HoSehBo 2

Contributions

Since we all did most of the tasks together, below are the contributions based on tasks rather than team members.
Format: **Task - Member 1** [most contributions *for this task*]**, [Member 2, Member 3], Member 4** [least contributed *for this task*] ### Members within brackets (2 & 3) have the same amount of contribution.

1. Algorithm for YouTube Hokkien Audio Collection (OCR) - Min
2. Website for in-person Hokkien Audio collection - Min
3. Sourcing of Hokkien Data (In person) - Alex, Habib, Min, Lim Jin Bin
4. Sourcing of Hokkien Data (YouTube) - Lim Jin Bin, Min, Habib, Alex
5. HTML & CSS - Lim Jin Bin, Alex
6. UI - Jin Bin, Alex, Habib, Min
7. Flask - Alex, Habib, Min, Jin Bin
8. API Integration - Habib, Min
9. Custom Model Training - Min
10. Slides & Report - [Min, Habib, Alex, Lim Jin Bin]
11. Video Shooting - [Min, Habib, Alex, Lim Jin Bin]
12. Video Editing - Min