AldiTalk

Project Proposal

T2
LIM JIN BIN [221128Z]
MOHAMMAD HABIB [231880L]
GERRELL ALEXANDER CHAN RUI BIN [230648A]
MIN PHYO THURA [233523A]

Table of Contents

Business Understanding

Company Background: Apex Harmony Lodge [Alex]

Overview

Apex Harmony Lodge (AHL), established in 1999, is Singapore's first purpose-built home dedicated to individuals with dementia. Their mission is to empower lives affected by dementia through strength-based, person-centred approaches in integrated dementia care. AHL offers both residential and community care services, aiming to reframe dementia as a unique phase of life worth celebrating.

Services Provided

- Residential Care: AHL offers various living arrangements tailored to the needs of residents:
    - Assisted Living (AL): Designed for individuals who are cognitively and physically able, promoting a vacationing, ability-centred lifestyle.
    - Supported Living (SL): For those requiring some assistance in daily activities, fostering an engaged community enabled by assistive devices.
    - Tender Loving Care (TLC): Focused on providing comfort and unconditional love to individuals in the later stages of dementia.
- Community Care: Beyond residential services, AHL extends support through programs like Therapy Through Work (TTW), EmbrACE, and Caregiver Mastery, aiming to integrate persons with dementia into the community and support caregivers.

Problem faced [Alex]

Possible Challenges Faced

Despite providing a multitude of services, it has been noted that AHL occasionally faces communication barriers between residents and caregivers, especially when language differences exist. Many residents speak languages or dialects which some staff may have trouble understanding, leading to potential misunderstandings and affecting the quality of care.

Proposed Solution

Implementing a speech-to-speech translation tool can bridge this communication gap, enabling caregivers to understand and respond to residents' needs more effectively. This tool would facilitate real-time translation between residents' native languages and the caregivers' language, enhancing interactions and ensuring that residents' preferences and concerns are accurately addressed.

Business Objective [Alex]

The main objective of this project is to improve the communication between residents of Apex Harmony Lodge (AHL) and their caregivers through the development of a speech-to-speech translation tool. This tool aims to bridge the language gap, allowing caregivers to understand and respond to the needs and preferences of residents who speak Mandarin, Malay, Tamil, or local dialects (Hokkien and Cantonese) accurately and efficiently. By reducing the downtime of communication, our solution will enhance the quality of care provided, align with AHL's mission of person-centered care, and improve the overall well-being and quality of life of both the residents and caregivers.

Data Mining Objectives [Min]

The data mining objectives for this project are focused on developing and optimizing the translation application. Key objectives include:

- Identify dominant and less represented languages/dialects among residents to prioritize S2ST development.
- Evaluate translation quality (e.g., BLEU scores) for supported languages and refine models for low-resource dialects like Hokkien.
- Analyze common translation errors in real caregiver-resident interactions, focusing on medical and emotional contexts.
- Test translation tool performance under various environmental noise levels to ensure reliability in real-world scenarios.

Data Understanding [Habib + Jin Bin]

Data

After conducting our initial research, we identified the most commonly used languages and dialects among the elderly in Singapore. This list includes Hokkien, Cantonese,Tamil, Malay, and Mandarin. While some of these languages are already supported by Azure AI's translation tools, others—especially the dialects such as Hokkien—face significant data scarcity.

Speech data for these dialects is not readily available in public repositories or linguistic datasets. To address this challenge, we plan to explore alternative methods for data collection, such as synthesizing speech data and transcribing old Hokkien dramas/ or other media content.

Data source

- Common Voice - Taiwanese Hokkien
- Chinese dialects (Hakka, Min Nan etc.) sentence pairs from Tatoeba

- Learn Hokkien Youtube videos
- Hokkien Dramas with Mandarin subtitles

Data Quality Assessment

- Completeness of the simulated data (e.g., coverage of dialects, edge cases).
- Accuracy and consistency in labels or translations.
- Noise levels or inaccuracies in simulated audio/text.
- Requires both the language/dialect and english voice/text recordings

Language and Domain-Specific Characteristics

- Challenges specific to each language (e.g., tonal variations in Mandarin, script complexity in Tamil).
- Domain relevance (e.g., inclusion of caregiving terminology).

Literature Review

Cascaded Speech-To-Speech Translation Using Base-Models[28] [Min]

Speech-To-Speech Translation (STST) systems traditionally involves three stages:

- Automatic Speech Recognition (ASR) model that takes in spoken speech as input and transcribes the text in the same source language,
- Machine Translation (MT) model that translates the text in the source language into the target language, and
- Text-To-Speech (TTS) model to generate the translated speech in the target language.

However, such an approach, where multiple models are stacked upon each other sequentially, significantly increases latency, and the error and noise from each model is compounded into the output, thereby limiting the overall performance.
In the chosen project from Hugging Face's Audio course, the developers used a two-step approach: cascaded Speech Translation (ST) and TTS model, which is robust and compute efficient than the traditional models.

ST

Whisper Base by OpenAI is used to directly translate the speech in the source language into text in target language, without the need to transcribe first and then translate. It has 74M parameters and can translate over 96 languages into English.

TTS

For generating translated speech in English, SpeechT5 TTS[29] is deployed throughout the process from tokenization of raw text to generating natural speech.

Technical details

'transformers' module is imported to load the Whisper Base and SpeechT5 models while PyTorch was used to distribute the computing load across local GPUs. A user's speech is inputted into Whisper Base, which then outputs the translated text in English. The text is processed by SpeechT5Processor and converted into mel-spectrogram by SpeechT5ForTextToSpeech, after which a high-quality speech audio is synthesised by the neural vocoder, SpeechT5HifiGan. IPython and Gradio are also used for recording, testing and playing the audio files.

Conclusion

Despite the straightforward implementation, this cascaded STST model offers reliable performance due to its state-of-the-art base models. However, it does not introduce any customization or fine-tuning, and the latency largely depends on the user's hardware system as the model is run locally.

Azure AI Integration[30]

Although the demo-model is developed locally using Hugging Face's *transformers* module, this approach of using base models is suitable for integrating with Azure AI services, which is also quite relevant to our project. Azure AI offers 1814 pre-trained AI models including Whisper and SpeechT5. Moreover, Azure AI allows for further fine-tuning and transfer learning approaches for developers, and it provides a cloud-based scalable architecture.
Implementing S2ST in Azure AI requires the traditional three-step approach: *Azure Custom Speech* for robust ASR, *Azure Custom Translator* for MT, and *Azure Custom Neural* Voice for Singlish-like ascent. These services are available with no-code usage at Azure Cognitive Services, but developing with Azure SDK in python provides higher accessibility and customization for our project.

ASR from Scratch[31] [Min]

There is a rational ground as to why researchers in S2ST try to by-pass the intermediate transcribing step; STT or ASR in itself is a complex model that requires considerable computing resources. This article will focus on building an ASR from scratch with SpeechBrain.

Data Augmentation and Feature Extraction

On top of noises in training data and real-world applications, audio data is difficult to gather, and thus, augmenting the data before training is a logical approach. Speech augmentation in SpeechBrain[32] is built mainly using PyTorch, Torchaudio, Math, Scipy, and NumPy. Some augmentation techniques are:

- Speed Perturbation, which alters the audio speed to either faster or slower,
- Time Dropout, which randomly removes chunks of the audio input to force the model to generalize the missing part,

- Frequency Dropout, which is similar to Time Dropout but certain frequencies instead of time frames are removed, and many more.

These augmentation algorithms can be applied in one step using the Augmenter method from SpeechBrain, thereby effectively creating more training data that is less prone to real-world adversaries.
For feature extraction, one way is to throw the data into Convolutional Neural Networks (CNNs) like in image processing, but SpeechBrain relies on known proper speech features[33, 34] such as Filter BANK (FBANK) and Mel-Frequency Cepstral Coefficients (MFCCs).

Acoustic Model

An acoustic model, or a speech recognizer, takes in speech data and outputs transcribed text. One of the acoustic models in SpeechBrain is a transformer-based encoder-decoder model. The encoder is based on CRDNN:

- Convolutional Neural Networks, which captures local and hierarchical patterns in features,
- Recurrent Neural Networks, specifically Long Short-Term Memory (LSTM), to model temporal dependencies in sequential speech input over time, and
- Fully-connected Deep Neural Networks to project the RNN outputs into desired dimensions.

The decoder then uses these hidden states, or outputs, from the encoder to predict the individual words based on attention-based Gate Recurrent Unit (GRU) with 1024 dimensions. The predicted words are evaluated by Cross Entropy Loss after the Softmax Activation layer.

Linguistic Model

The acoustic model transcribes word by word and does not consider the semantic meaning of the whole outputted sentence, which may not perform well in most cases. Words with similar pronunciation such as ‘read’ in past tense and ‘red’ can hinder the accuracy. Instead of naively taking every word with the highest probability, a linguistic model is used to evaluate the probability of the whole sentence produced from the acoustic model. Doing so will result in the sentence “I read a book” having higher probability than “I red a book”, thereby increasing model performance.
This is done in SpeechBrain using BeamSearch with CTC loss (Connectionist Temporal Classification).

Conclusion

Such an extensive model would be invaluable in picking up user voices with varying physical and acoustic properties in a stand-alone STT model, but in S2ST, this approach is too computationally taxing to be followed by two other MT and TTS models; the latency is noticeable. This is one of the reasons why new research is being done to directly translate speech to speech without any intermediate steps like text or discrete units.

Direct Speech-To-Speech (STS) Translation with a Sequence-To-Sequence Model[35] [Min]

This research is one of the attempts in trying to by-pass the intermediate layers in S2ST by directly translating source speech to target speech. It introduces the Translatoron, which is built on Tacotron2 TTS. The model directly converts the input mel-spectrogram in a source language (Spanish) into a matching mel-spectrogram in the target language (English), generating the translated audio.

Implementation

The main idea behind this is quite simple: to use matching input/output speech pairs as training data. Yet, the actual data collection is much more demanding compared to text-to-text pairs in MT and speech-to-text pairs in STT. After collecting 1527 hours of Spanish speech data and synthesizing 715 hours of it in English, the training spectrograms are inputted into the encoder with 8 layers of stacked Bidirectional LSTM (BLSTM) for learning sequential temporal dependencies, followed by a multi-head additive self-attention layer with 4 heads. The decoder then takes in these encoded hidden states and generates the translated sequence of log-mel spectrograms in target language, which produces the audible translated speech using the WaveRNN neural vocoder.

Conclusion

Although the training data size is limited, the research proved the feasibility and potential of direct S2ST. Translatoron performed slightly less accurately compared to a tradition cascaded S2ST, but it was able to retain non-linguistic information such as emotion and prosody in spoken speech.

Speech-to-Speech Translation for a Real-World Unwritten Language[36]

This study explores the development of a speech-to-speech translation (S2ST) system for translating between English and Taiwanese Hokkien. It talks about the challenges related to data scarcity and the lack of standardized text writing systems.

Data challenges: [Jin Bin]

- Hokkien lacks a standardised writing system, with limited availability of transcriptions or annotated datasets. This posed a significant barrier to training effective models.

Data solutions: [Jin Bin]

1. Human annotation: Mandarin was used as a pivot language because there weren't many bilingual speakers who could directly translate English to Hokkien. For the Hokkien→English translation, they used Hokkien dramas with Mandarin subtitles and had them translated into English by bilingual speakers.

They also utilized a dataset called Taiwanese Across Taiwan (TAT), which contained Hokkien speech and the corresponding transcripts and had those translated directly into English. For the English→Hokkien translation, the researchers took Mandarin text from the MuST-C dataset and had it translated into Hokkien.

2. Mined data: Leveraged machine learning techniques to extract Hokkien speech data and align it with English or Mandarin translations from unlabeled sources.
3. Weakly supervised data: Used pseudo-labeling techniques to produce weakly supervised datasets, bridging the gap in data resources

Models: [Jin Bin]

The project utilized two main speech-to-speech translation (S2ST) models: the single-pass decoding S2UT and the two-pass S2ST system (UnitY). The single-pass decoding S2UT model directly translates source speech to target speech, bypassing intermediate text generation. This approach is simpler and more efficient, reducing complexity and minimizing potential error propagation. However, it performs poorly in low-resource settings, particularly for distant language pairs, and achieves very low BLEU scores in both translation directions when trained solely on human-annotated data. On the other hand, the two-pass S2ST system integrates an intermediate step that leverages Mandarin text for additional supervision, significantly improving translation quality. This model consistently outperformed the single-pass system, achieving higher BLEU scores in both English-to-Hokkien and Hokkien-to-English tasks. The use of a high-resource pivot language like Mandarin proved beneficial for enhancing performance with low-resource languages such as Hokkien. However, this approach is more complex, requiring additional text prediction steps and greater computational resources.

Conclusion: [Jin Bin]

While the single-pass model is simpler and efficient, the UnitY model consistently outperformed it in both English to Hokkien and Hokkien to English translation tasks**.** This suggests that leveraging additional text supervision from a similar language like Mandarin is beneficial for improving S2ST performance, especially for unwritten languages like Hokkien. Therefore, given our similar position of having a lack of data, we could employ this method for our hokkien translation.

Project Solution Approach [Min]

AldiTalk, at its core, has three main functionalities: transcribing the speech to text, translating it, and reading it out loud. But, with the target user group being Singapore

senior citizens, there are a number of other factors affecting the project solution approach such as:

- Local dialects
- Languages with limited resources, and
- Elderly user friendliness.

Only when these factors are fully incorporated with the three main functionalities, should the AldiTalk project be deemed successful.

Phase 1: EDA for Local Dialects

Singaporeans are multilingual people; over 74% of the population can speak two or more languages according to Singapore Census of Population 2020[37]. While English, Mandarin, Malay and Tamil are dominant, there are also many other languages and dialects spoken among the residents of Singapore, especially among senior citizens. For AldiTalk to be effective and useful for its target users, it is necessary to comprehensively analyze the language proportions among Singapore elderly and integrate them accordingly.

Exploratory Data Analysis will be done using *Python and Matplot* to find out the most common dialects spoken among Singapore senior citizens and determine which of them to focus on for AldiTalk to have the greatest impact.

Phase 2: Data Sourcing & Augmentation

Apart from the four official languages in Singapore, resources for other dialects are quite limited in terms of both quantity and diversity. Since an AI model can perform only as good as the data provided to it, sourcing quality data of necessary size is crucial. Data sources provided in *Data Understanding* are highly unlabelled and require manual work. Audio augmentation techniques mentioned in *ASR from Scratch* will be applied here as well.

Audacity for recording English translation, Aneas for annotating, and other usual Python libraries such as IPython, Torchaudio, Librosa, and SpeechBrain will be used in preparing and preprocessing the data.

Phase 3: Model Training & Fine-tuning

Since direct S2ST is still in the research and development times, there are two ways to implement AldiTalk with pre-trained base models:

- Two-step approach with speech translation and synthesizing translated speech is a preferred method and is similar to the cascaded method described earlier. Fine-tuned Whisper STT Translation[38] is used to eliminate the intermediate transcription, and Azure AI Speech will generate audible translated speech in English. This method is computationally faster than the three-step approach but has limited api requests per minute and is expensive.
- The cascaded three-step model may be slightly slower compared to direct translation but is logical and always offers better results. In addition, each step

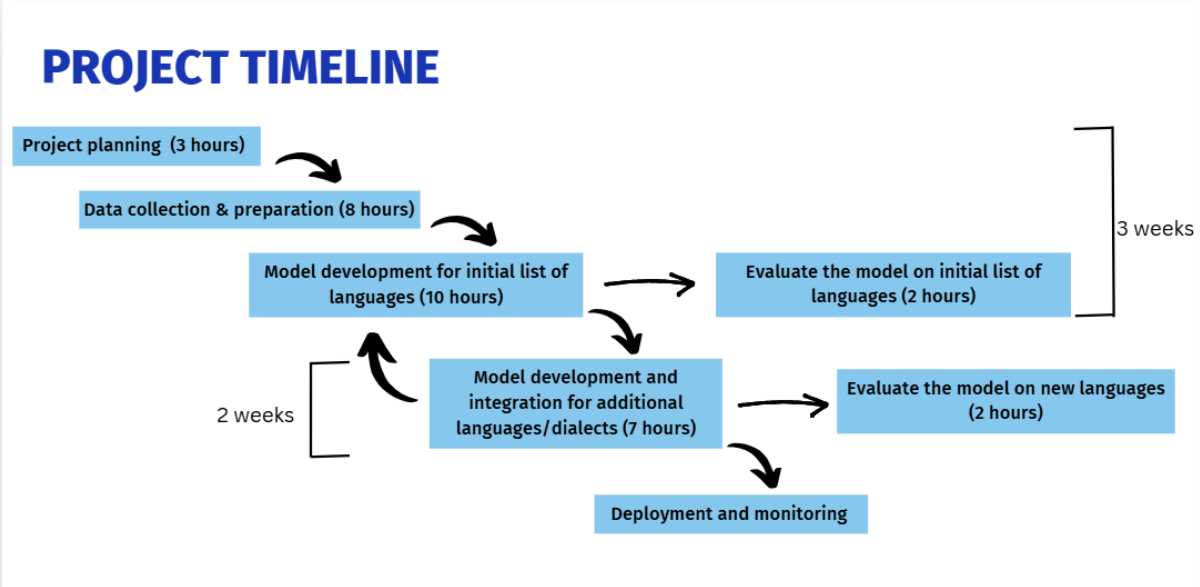
of ASR, MT, and TTS can be adjusted to tailor-make the model to our specific needs. Robust ASR will be implemented from scratch (if possible, if not Azure Custom Speech) using several techniques in dealing languages with limited resources[39]. Azure Custom Translator for MT and Custom Neural Voice for TTS will be fine-tuned and implemented.

Addressing physical and linguistic noise in real-world applications and incorporating medical vocabulary for use in nursing homes will be considered as well. Both approaches will be tried out first, and the doable with better performance and efficiency will be introduced to the users.

Phase 4: Elderly User Friendliness

It is undeniable that the pace of civilization has been speeding up since the development of the internet, leaving those who cannot keep up behind; most elderly are a part of those being left behind, and therefore, for an application targeted to improve the communication between care-givers and old people, possibly diagnosed with dementia, user friendliness plays a big role in its success and applicability. Dart programming language with Flutter framework is a great choice for AldiTalk as it is a cross-platform framework with an impressive UI and provides fast performance comparable to native languages like JavaScript.

Project Plan (Incremental Project Lifecycle) [Jin Bin]



Project planning:

- Objective: Clearly define project goals, scope and stakeholders
- Task:
  - Identify the target languages/dialects
  - Conduct research on any existing translation models
  - Prepare a project timeline
- Deliverables: Timeline

Data collection & preparation:

- Objective: Gather the necessary data and preprocess it for training the translation model
- Task:
    - Look for conversational datasets in the targeted languages that are not currently in Azure AI (focus on caregiving scenarios)
    - If language/dialect is low resource language, find alternatives like transcription of media content.
    - Clean and preprocess data
- Deliverables: Preprocessed dataset

Model development for initial languages:

- Objective: create the translation tool with the basic languages provided by Azure AI
- Task:
    - Develop a prototype by focusing on speech-to-text, text translation and text-to-speech for existing languages in Azure AI first
- Deliverables: Prototype translation tool

Evaluation for initial languages:

- Objective: Validate the model and to improve its accuracy
- Task:
    - Conduct tests on the model
    - Identify any issues (e.g cultural nuances, translation speed)
    - Refine the model
- Deliverables: Understanding the issues so we can fix it and prevent the similar things from happening during the training of other languages

Model development for additional languages:

- Objective: integrate dialects like Hokkien that were not originally provided by Azure A into the model
- Task:
    - Train the model using the data found
    - Conduct similar validation steps as with the initial model to ensure quality and robustness.
- Deliverables: Understanding the issues so we can fix it and prevent the similar things from happening during the training of other languages

Deployment

- Objective: prepare for live demonstration of model
- Task:
    - Develop presentation materials (slides, demos, scripts).
    - Showcase the model's capabilities with real-time demonstrations.

We will also meet up face to face or through discord calls for about an hour every 2 days to discuss our progress and any problems encountered.

References [Habib]

1 Microsoft. (n.d.). *AI Translator: Accurately translate text in more than 100 languages*. Azure AI Services.

2 Hutchins, J., & MT News International. (1999). *Warren Weaver memorandum, July 1949*. MT News International, (22), 5–6, 15.

3 Hutchins, J. (2006). *The first public demonstration of machine translation: The Georgetown-IBM system, 7th January 1954*.

4 Chomsky, N. (1956). *Three models for the description of language*. Cambridge, MA: Department of Modern Languages and Research Laboratory of Electronics, Massachusetts Institute of Technology.

5 Intel Corporation. (n.d.). *The story of the Intel 4004*.

6 Freiberger, P. A., Hemmendinger, D., Pottenger, W. M., Swaine, M. R., & The Editors of Encyclopaedia Britannica. (n.d.). *The personal computer revolution*. Encyclopaedia Britannica.

7 Mondal, S., Singh, R., & Kulkarni, A. (2023). *Evolution of storage technology: From punch cards to cloud computing*. *International Journal of Creative Research Thoughts (IJCRT), 11*(9), 266–269.

8 Shannon, C. E. (1948). *A mathematical theory of communication*. *The Bell System Technical Journal, 27*(3), 379–423; 623–656.

9 Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). *Building a large annotated corpus of English: The Penn Treebank*. *Computational Linguistics, 19*(2), 313–330.

10 Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). *A statistical approach to machine translation*. *Computational Linguistics, 16*(2), 79–85.

11 USC Information Sciences Institute. (2001). *Hansards dataset*. Hugging Face.

12 Church, K. W., & Hanks, P. (1989). *Word association norms, mutual information, and lexicography*. *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics (ACL)*, 76–83.

13 Collins, M. (n.d.). *Statistical machine translation: IBM models 1 and 2*. Annotated lecture notes.

14 Loper, E., & Bird, S. (2002). *NLTK: The Natural Language Toolkit*. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Association for Computational Linguistics.

15 Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). *BLEU: A method for automatic evaluation of machine translation*. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 311–318). Association for Computational Linguistics.

16 Se, K. (2024, May 16). *The recipe for an AI revolution: How ImageNet, AlexNet and GPUs changed AI forever*. Turing Post.

17 Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*.
18 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*.
19 Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*.
20 Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018, June 11). *Improving language understanding by generative pre-training*. OpenAI.
21 Specialty Answering Service. (n.d.). *The Voder: First human speech synthesizer*.
22 Lewis, R. (2024, November 20). *Auld Lang Syne*. In *Encyclopaedia Britannica*.
23 Moskvitch, K. (2017, February 14). *The machines that learned to listen*. BBC Future.
24 van den Oord, A., & Dieleman, S. (2016, September 8). *WaveNet: A generative model for raw audio*. DeepMind.
25 Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y., & Wu, Y. (2018). *Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions*.
26 Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). *Robust speech recognition via large-scale weak supervision*.
27 3Blue1Brown. (2024, Nov 20). Large Language Models explained briefly. YouTube.
28 Hugging Face. (2023, July 12). *Speech-to-speech translation*. Hugging Face.
29 Ao, J., Wang, R., Zhou, L., Wang, C., Ren, S., Wu, Y., Liu, S., Ko, T., Li, Q., Zhang, Y., Wei, Z., Qian, Y., Li, J., & Wei, F. (2022). *SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing*.

30 Bisson, S., Branscombe, M., Hoder, C., & Raman, A. (2022). *Azure AI services at scale for cloud, mobile, and edge*. O'Reilly Media.

31 Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., De Mori, R., & Bengio, Y. (2021). *SpeechBrain: ASRfromScratch.ipynb* [Notebook]. Google Colab.

32 SpeechBrain. (2023). *Augmentation toolkit* (Commit ID: 738ffae) [Source code]. GitHub.
33 Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., De Mori, R., & Bengio, Y. (2021). *SpeechBrain: FourierTransform+Specgrams.ipynb* [Notebook]. Google Colab.

34 Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., De Mori, R., & Bengio, Y. (2021). *SpeechBrain: SpeechFeatures.ipynb* [Notebook]. Google Colab.

35 Jia, Y., Weiss, R. J., Biadsy, F., Macherey, W., Johnson, M., Chen, Z., & Wu, Y. (2019). *Direct speech-to-speech translation with a sequence-to-sequence model*.

36 Chen, P.-J., Tran, K., Yang, Y., Du, J., Kao, J., Chung, Y.-A., Tomasello, P., Duquenne, P.-A., Schwenk, H., Gong, H., Inaguma, H., Popuri, S., Wang, C., Pino, J., Hsu, W.-N., & Lee, A. (2022). *Direct speech-to-speech translation for a real-world unwritten language*.

37 Singapore Department of Statistics. (2020). Census of population 2020: Statistical release 1 - Infographics.

38 HueiShuang. (2024). Chinese/Taiwanese Whisper ASR Project. GitHub.

39 Kumar, L. A., Renuka, D. K., Chakravarthi, B. R., & Mandl, T. (2024). Automatic speech recognition and translation for low resource languages. Wiley-Scrivener.

Appendix

History of Machine Translation [Min]

Introduction

Modern translation services such as Azure Speech Translation can seamlessly translate between over 100 languages, either in text-to-text or direct speech-to-speech translation.[1] Such services are available at a click of a simple user interface (UI), but the effort behind it is no easy feat, rather a pinnacle of humans, especially that of Natural Language Processing (NLP) scientists. NLP has evolved dramatically since the early 1950s, when early pioneers tried to hard-code all the words and grammatical rules of a language one by one.[3] This literature review will explore the advancements in NLP, adversaries in each era and how they were overcome, with a focus on Machine Translation (MT).

Era of Hard-Coded Models

It is undeniable that the world of NLP evolved around machine translation. Since the start of computers with punching cards and binary systems, people have long dreamed that machines would one day, despite the skepticism, understand natural language and be able to translate from one to another.[2]
It was the 1954 George-IBM Machine Translation Demonstration[3] that shattered such skepticism and sparked the positive interest in NLP and MT. 250 Russian words were manually punched onto cards as the base lexicon, or vocabulary. The input Russian sentence was then broken down into individual words, translated into English word by word and assembled into a sentence using 6 pre-defined syntax rules. The experiment was a sensational start towards today's NLP systems, but it showed some fatal flaws of limited scope and lacked the morphological analysis of the language.
The latter challenge was tackled two years later in 1956 by Noan Komsky. He proposed Three Models for the Description of Language[4]: Finite State Markov Processes, Phrase Structure Grammars, and Transformational Grammars. These three models complemented each other in understanding and generating complex English sentences by separating them from grammatical transformations (which introduced complexity) into simple basic sentences. As the research was primarily focused on grammatical structures, the main disadvantage of this approach was that it did not learn from data.

Statistical and Machine Learning Models

Transition from rule-based language understanding to statistical approach was a big leap towards today's NLP algorithms. It took over 3 decades to address the challenges in statistical language models: computational power, storage capacity,

digitized data. Until the first commercial microprocessor, Intel 4004[5], was released in 1971, there was no central computing unit, and computers were bulky due to vacuum tubes, transistors and circuits-based systems[6]. The computing power was quite limited for statistical programming. Language data was often in paper form, not digitized; furthermore, storage technologies using punch cards and magnetic tapes[7] were insufficient for managing large language data.

On top of advancement in computing hardwares, it was largely due to Claude Shannon's Mathematical Theory of Communication[8], and large annotated language datasets like the Penn Treebank[9] that statistical application in language processing became feasible.

In the Statistical Approach to Machine Translation paper by Peter F. Brown[10], he explored the idea that every sentence in one language is a possible translation of any sentence in the other language. Using Canadian Hansards dataset[11], the experiment translated Russian sentences into English with reasonable translation for about half of the sentences. It used an n-gram model, which estimated the word based on the preceding n-1 words, and assigned probabilities to every possible sentence, where the highest one is chosen as the correct translation, based on Naive Bayes theorem. Despite the model not learning the actual meaning of the words, it was able to generate acceptable translations, purely based on statistics.

In another research, called Word Association Norms, Mutual Information, and Lexicography[12], the researchers used a bi-gram model to interpret the semantic relationships of two words. They introduced the 'association ratio', popularly known as Pointwise Mutual Information (PMI), which is a measure of how often two words co-occur within a certain window of words in a corpus of text, effectively learning synonyms, and antonyms, etc.. Words like "set" and "off" often occurred together, thereby providing "set off" as a meaningful word to include in dictionaries. Others like "Doctor" and "Nurse" also showed high association ration, possibly accounting for their similar context. Nonetheless, the model lacked to consider the syntactic structure of sentences, highlighting the need for other supportive methods.

IBM Model 1 and IBM Model 2 (IBM-M2) were two notable machine translation models based on pure statistics, with an introduction to Machine Learning (ML)[13]. Both models used the noisy channel approach, assuming that an original French sentence was the result of the English sentence being corrupted, or distorted, due to noise. The task of the MT model is to recover the correct English sentence through conditional probabilities. This approach introduced ML in that there were learnable translation and alignment parameters, and that the model improved after iterative refinement. Moreover, the ultimate goal of IBM-M2 was to generalize from training data so that the model performed well on unseen new French sentences.

Later in the early 2000s, Natural Language ToolKit (NLTK) was released[14], compiling all previous NLP algorithms into a modular, simple and highly efficient python library. Some commonly used methods are tokenization, POS tagging, and corpus handling. A number of translation and NLP researches popped up afterwards, and Bilingual Evaluation Understudy (BELU) was a key player in evaluating these models, allowing for rapid testing and refinement of new modeling ideas[15].

Deep Learning

When the GPUs-based deep learning neural networks gained popularity in 2012 (mainly due to the AlexNet)[16], researchers tried throwing existing statistical corpus data into Neural Network Language Models (NNLMs) and Recurrent Neural Network Language Models (RNNLMs) to test the capabilities of deep learning in NLP. Despite outperforming existing ML models, computing complexity and data scarcity posed significant challenges for advancement.

As such, Efficient Estimation of Word Representations in Vector Space by Google was a major breakthrough, focusing on learning high-quality word vectors from large text datasets[17]. The research proposed two models: Continuous Bag-Of-Words (CBOW) and Continuous Skip-Gram. Unlike usual neural networks, the models did not have any nonlinear hidden layers but focused on linear regularities of larger data volume, thereby improving simplicity, efficiency, and scalability. It was also the first time that semantic-syntactic word relations were introduced, such as “Paris - France + Germany = Berlin.” The output word vectors had many applications beyond simple word similarity tasks; they were later applied in machine translation, sentiment analysis, and knowledge-based completion etc.

Another recent major breakthrough in NLP was the birth of transformer architecture. In the famous “Attention is All You Need” paper[18], researchers from Google came up with a self-attention mechanism with positional encoding, which tells the model to focus on a specific part, or words, of a sentence in predicting the next word. It offered incredibly efficient parallel computing with long-range dependencies for language understanding, as well as the intuitive interpretability of the use of ‘attention’. Some state-of-the-art models based on transformers are BERT(Bidirectional Encoder Representations for Transformers), which is a bidirectional encoder transformer[19], and OpenAI GPT (Generative Pretrained Transformers), a unidirectional decoder transformer[20]. The former excels in coherent contextual understanding while the latter is used for text generation and chat completion.

Speech Integration

Analyzing and synthesizing audio data has also evolved alongside NLP. The first ever electronic voice synthesizer can be traced back to 1939: the voice operation demonstrator, or Voder for short[21]. Mimicking the human vocal tract, the Voder operated similarly to a piano keyboard and was able to transmit recognizable human speech. It could even stress specific words of a phrase or sing Auld Lang Syne[22]. From the same company that created the Voder, Audrey was also introduced as the first automated speech recognition system that recognized spoken digits from zero to nine with over 90% accuracy. It was followed by Harpy in 1976 from Carnegie Mellon University, which could understand up to 1,011 English words. Surprisingly, these models only utilized simple audio signal matching and linguistic modeling. The first commercial speech recognition software was released in 1990 known as Dragon

Dictate, which used statistical models like Hidden Markov Models and was priced at hefty $9,000[23].
Fast forward, today's text-to-speech (TTS) and speech-to-text (STT) systems are based on deep learning, specifically Recurrent Neural Networks (RNNs) and transformers. Some of the notable models are Google's WaveNet[24] and Tacotron 2[25], and OpenAI's Whisper[26].

Transfer Learning & Fine-Tuning

Whisper bwy OpenAI provides state-of-the-art STT translation but is trained on a massive dataset of 680,000 hours of multilingual and multitask supervised data collected from the internet[26]. Although not built for machine translation, GPT3 also performs quite well in translating texts with zero-shot learning, but it is a Large Language Model (LLM) whose training corpus is so large that an average human would need 2600 years even if he read 24-7[27]. This is not the amount of resources that an average company can invest in.
Thus, most developers build their models on existing ones through transfer learning, and fine-tuning. This approach tests out the base open-source model for the targeted purpose like local dialect translation to determine its performance. The model is then further adjusted on a much smaller specific dialect dataset and fine-tunes the parameters, which considerably reduces the resources required in developing translation services. Some of these approaches are discussed in detail as they relate the most to our desired project.