DLSU Research Congress 2025
De La Salle University, Manila, Philippines
June 26 to 28, 2025

Development of ASR models for a Filipino healthcare chatbot

Matthew Chua [1], Vincent Burce [2], Ashley Ramos[3], Merrick Malong [4], and Judith Azcarraga[1,*]

[1] *De La Salle University*
matthew_adrian_u _chua@dlsu.edu.ph
vincent_burce@dlsu.edu.ph
ashley_kylle_ramos@dlsu.edu.ph
merrick_malong@dlsu.edu.ph
judith.azcarraga@dlsu.edu.ph

Abstract: In the Philippines, the availability of accurate Automatic Speech Recognition (ASR) systems remains limited. This is especially true for local languages like Filipino and in domains such as healthcare, where speech-based technologies can significantly improve communication and access to information. This gap becomes more pronounced when it comes to children's speech, which presents unique challenges due to age-related variations in pronunciation, pacing, and intonation. This study explores the development of an Automatic Speech Recognition (ASR) system tailored for Filipino children's speech, specifically Tagalog. Different speech models were built using female adult and children's speech data to determine the most suitable model for the available data. The best performing ASR model will be integrated into FilBis chatbot, a wellness monitoring system for young children. An adult speech dataset from a related study, along with a newly gathered dataset of children's speech, are used as the primary training data for the improved model. Both datasets consist of audio recordings of speech read aloud from FilBis. The best performing ASR model will be integrated into Filbis chatbot, a wellness monitoring system for young children. Although initial testing showed promise, the best-performing model only surpassed the baseline when trained on sufficient data, with a Word Error Rate of 4.15% compared to the baseline's 9.88%. This highlights the need for more high-quality, annotated speech datasets from Filipino children.

Key Words: Filipino; Automatic speech recognition; Artificial intelligence; Health informatics; Natural language processing

1. INTRODUCTION

Automatic speech recognition (ASR) systems have advanced significantly in recent years, particularly for high-resource languages with abundant transcribed data, such as English and Mandarin (Radford et al., 2022). However, languages like Filipino remain underrepresented in ASR research due to a lack of high-quality, domain-specific linguistic resources. This scarcity is especially problematic in healthcare, where

ASR plays a growing role in applications such as medical transcription, telemedicine, and voice-assisted diagnostics. The limited availability of standardized Filipino speech datasets poses a major barrier to developing accurate and culturally relevant ASR systems for Filipino-speaking populations, highlighting the urgent need for focused research and dataset development in this field (Montalan et al., 2025; FlipStar, 2025).

Additionally, developing speech recognition systems for children presents unique challenges compared to systems designed for adults. Children's speech differs significantly in terms of pitch, pronunciation, articulation, and fluency, often exhibiting higher variability and less predictable patterns. These differences can reduce the performance of ASR systems trained primarily on adult speech, leading to higher word error rates when applied to children's voices. In addition, the availability of large-scale, high-quality annotated speech datasets for children is limited, especially for low-resource languages like Filipino, further complicating model training and evaluation (Shivakumar et al., 2020).

An intersection between these two problems in the field is the lack of accessible healthcare resources for Filipino public school children. Medical facilities are often underdeveloped, especially in rural areas of the country. This makes it difficult for elementary-aged children to seek medical care when they are experiencing symptoms of an illness, even with the support of their parents.

With this, the goal of this research is to develop an Automatic Speech Recognition (ASR) system targeted towards young children with an emphasis on health-related contexts. Different ASR models are explored in the study to compare their performance with the given data. A mix of both adult and children's speech data is used to train the resulting model. The improved model will then be implemented into FilBis, a children's healthcare chatbot.

The findings of the study are anticipated to benefit the field of computer science - specifically language models that aim to have Filipino speech as input. Additionally, the field of healthcare will also benefit from the results. The improvement of speech recognition catering towards specific health-related keywords may assist healthcare professionals by making technology such as healthcare chatbots more reliable.

2. METHODOLOGY

One of the two speech corpuses used in the study comprised female adult voice actresses reading health-related words aloud. Studies have shown that female voices are the closest in effect to children's voices, which are typically higher pitched. A second corpus of actual children's speech was gathered at an elementary school in Cavite. This second corpus consisted of speech from 5 different speakers. The total length of segmented Filipino audio for the children's and adult's speech data is approximately 2 hours and 9 hours, respectively.

To address the challenges of developing an automatic speech recognition (ASR) system for Filipino health-related speech, this study explores a range of state-of-the-art models and tools, each with distinct architectures and strengths. The following ASR approaches were evaluated: Google Web Speech API, Time Delay Neural Networks (TDNN), a Hybrid CTC-Attention model, Meta's XLS-R 300M model, OpenAI's Whisper model, and a fine-tuned version of Whisper. These models were selected based on their performance in prior related works across various speech tasks. For instance, Google's Web Speech API is the initial model used by the FilBis Chatbot, while TDNNs have shown success in low-resource and speaker-independent ASR tasks (Pascual. et al., 2023). Meta's XLS-R model builds on the success of multilingual representation learning from Wav2Vec 2.0 (Baevski et al., 2020), and Whisper's multilingual and multitask learning capability has demonstrated robustness in noisy environments (Radford et al., 2022). Each subsection below elaborates on the

implementation, training strategy, and evaluation of these models as applied to Filipino medical speech, with citations to related research where applicable.

2.1 Google Web Speech API

The best performing ASR model will be integrated into the Filbis chatbot, a wellness-monitoring chatbot that asks a series of questions, simulating a nurse or a doctor asking questions about the different body systems of a child. A key feature of FilBis is its speech recognition system that allows the user to verbalize their answers instead of having to click on-screen buttons or type in a text field.

The baseline model that is currently implemented in the FilBis chatbot utilizes Google's Web Speech API, a browser-based system that allows for quick integration of speech recognition and synthesis in web applications (MozDevNet, n.d.)

Web Speech API was chosen as the ASR system for FilBis as it was free and also fairly powerful. It was, for the most part, accurately able to detect Filipino speech - albeit, this was not tested on children's voices.

Aside from the Web Speech API built into the browser, FilBis also has a somewhat crude implementation of 'synonyms' and 'ignore words' which help the application map transcribed speech to the available options of whatever the current question is. It is worth noting, though, that this system is reliant on the fact that the recognized speech is an accurate transcription of the input spoken. Hence, it would be an improvement to the system as a whole if a more fine-tuned model were to replace the Web Speech API.

2.2 Time Delay Neural Networks

Time delay neural networks were implemented using Python and Tensorflow library, which uses multiple 1D convolutional layers. Unlike a normal 2D convolutional neural network, by having only one dimension, this allows the model to apply filters along the single dimension over time capturing patterns and features across different time steps. After each convolutional layer, they are stacked with each other to capture longer temporal dependencies.

The specifics of the implementation are as follows: initially, it uses MFCC to extract the features of the speech audio, after which a label is placed onto the MFCC values corresponding to the word mappings in the dictionary. It uses a dataset with about 10000 female-recorded segmented audio. The dataset uses a 85%-15% split for training and testing the model. Additionally, the data for the test set uses a speaker that was not used in the training data to ensure its validation accuracy on unheard voices.

The model uses 3 hidden 1D convolutional layers and 1 dense connected hidden layer at the end. For each layer, ReLu activation and padding of "same" was used. A Dilation rate of 1, 2, and 4, respectively was used for each layer. This allows the model to capture both short-term and long-term contexts. Additionally, bidirectional LSTM was used, allowing the model to further track dependencies across time from both the past and future contexts.

Batch normalization was used for faster convergence and avoiding overfitting. Finally, Adam optimizer was used for its adaptive learning rate and since the model is trained on MFCC features, it helps reduce noise that makes the gradient updates more stable.

2.3 Hybrid CTC-Attention Model

The initial approach in implementing a Hybrid CTC-Attention model was through a combination of using ESPNet in order to set up and train the model and Kaldi in order to preprocess the audio dataset and transform it into a suitable format that the model can use and recognize for training. However, due to running

into significant challenges and difficulties in implementing the software, Python was used.

The implementation in Python utilizes character-level for its tokenization. In terms of preprocessing, Mel Spectrogram was used in order to extract the features from the audio files. With the nature of the Hybrid CTC-Attention, the loss function is a combination of the Connectionist Temporal Classification and the Attention for improved performance. The weights for each ~~of them~~ can also be adjusted. As for training and testing, similar to the implementation in the TDNN model, the training-testing split was set to 85/15 to ensure consistency across the board. During training as well, various hyperparameters were modified in order to attempt to find a suitable combination that provided satisfactory performance. The model, being set up from scratch, means that it was not pretrained on any other data. Evaluation of the performance was done through the WER. The final results from adjusting the hyperparameters are discussed in the following section.

2.4 Meta's XLS-R 300M Model

Wav2Vec 2.0, released by Meta in 2020, is a self-supervised learning model that is built upon the original Wav2Vec framework introduced in 2019. It is a transformer-based speech recognition model, with its feature encoder being CNN and the transformer encoder operating similar to Bidirectional Encoder Representations from Transformers (BERT). An extension from this model is XSL-R, which is Meta's cross-lingual approach (Baevski et al., 2020).
The model is pre-trained on 436k hours of unlabelled speech, that of which includes VoxPopuli, MLS, CommonVoice, Babel, and VoxLingua2017. The speech data covers 128 languages, Filipino included (AI at Meta, n.d.).

There are various versions of the model, with the differences coming in the number of parameters. The versions are 300M, 1B, and 2B. For the purposes of evaluation, the 300M would be explored and evaluated. The model was trained and evaluated on the same dataset as the TDNN model.

2.5 Fine-tuned XLS-R 300M model

The model was fine-tuned and evaluated with the dataset used in the previous implementation with TDNN. The dataset was formatted as JSON with references to their local audio files and their transcriptions. The audio was preprocessed using the Wav2Vec2FeatureExtractor in order to normalize the waveform impetus. The transcriptions were then tokenized, which was created from the dataset beforehand. Afterwards it was then loaded through the Wav2Vec2CTCTokenizer. For the sake of consistency, the audio samples were converted to mono as well.

For training, the model was trained with the Trainer API from the Hugging Face Transformers with the AdamW optimizer. Its learning rate was set to 3e-5. It was trained for 10 epochs with a batch size of 8 and a gradient accumulation of 2 steps. The feature encoder of the model was also frozen in order to speed up training and prevent overfitting, given the amount of the dataset. After training, the final model is evaluated with the WER from jiwer.

2.6 OpenAI Whisper Model

OpenAI's Whisper is a transformer-based speech recognition model with an encoder-decoder architecture, commonly used in sequence-to-sequence models. The encoder takes in an audio input, which is first converted into a log-Mel spectrogram and processes it to create a learned internal representation. This representation is then passed to the decoder, which generates the transcribed text step-by-step. What makes Whisper unique is that it incorporates special tokens to control tasks like language identification and translation, allowing the model to support multilingual transcription, including Filipino (Radford et al., 2022).

In terms of implementation, Whisper was pre-trained on a large dataset composed of 680,000 hours of audio in multiple languages, although over 60% of the dataset is English (Radford et al., 2022). This introduces a known bias towards English, but also gives the model strong multilingual capabilities, making it suitable for our use case involving Filipino speech. It also performs well on long audio segments and noisy environments due to the robustness gained from its diverse training data.

For the setup, the medium version of Whisper was chosen as it strikes a good balance between performance and speed. The medium model consists of 24 transformer layers, each with a width of 1024 and 16 attention heads - amounting to roughly 769 million parameters (Radford et al., 2022). Compared to the small model which has 12 layers, the medium version offers better feature extraction and higher transcription accuracy, especially for domain-specific terms. While the large model has even more layers, it often overgeneralizes and consumes more resources, making the medium model a more efficient choice.

The model's transcription accuracy was evaluated using the Word Error Rate (WER) metric by comparing predicted outputs against manual hard-coded transcriptions, using the JiWER Python library. While not perfect, Whisper's performance with Filipino and noisy data made it a strong starting point for our speech recognition task.

2.7 Fine-tuned OpenAI Whisper Model

Fine-tuning the Whisper model involves continuing training from a pre-trained checkpoint using an audio and text dataset. The dataset is passed through Whisper's feature extractor to convert the audio into spectrograms. Which are then fed into the encoder-decoder transformer model, the decoder then learns to predict the target text (Gandhi, 2022).

The same dataset and process were used for training and testing as discussed in the TDNN section where the test data contains only 1 speaker whose voice was not used in the training to ensure non-overlapping train and test sets. The audio data is first converted into log-Mel spectrograms and its corresponding text transcription into a tokenized format, these are essentially its label IDs similar to the process in the TDNN model. After which a custom DataCollator was used to pad both the audio features and tokens to ensure all data are of the same length.

The model is then trained using a learning rate of 0.0001, batch size of 16 and a max steps of 5000, after every 1000 steps a checkpoint is saved to evaluate the best performing model across the entire training sequence. 5000 was chosen as the max step value as when conducting initial training beyond the 3000 steps mark, the loss and learning rate difference in subsequent steps yielded insignificant gains or improvements. After every 1000 steps the model is evaluated using the WER formula by leveraging the Jiwer library

Training on a medium-sized Whisper model was also attempted however, due to hardware limitations, the training sequence was not conducted since the estimated training time was 231 hours and rendered the computers unusable during the training process.

3. RESULTS AND DISCUSSION

To evaluate the best performing model in the experiments, the word error rate was calculated using Equation 1, where I is the number of insertions, D is the number of deletions and S is the number of substitutions. Then the sum is divided by the total number of words N.

$$WER = (I + D + S) / N \quad \text{(Eq. 1)}$$

where:

I = number of insertions
D = number of deletions
S = number of substitutions

N = total number of words

Initial testing of the models was first conducted with the adult speech dataset. This is due to the adult dataset being much larger and of a better quality than the children's dataset. Models that are not performing well in the initial testing most likely will not be suitable for use with children's speech. A total of 1336 segmented audio files were used for testing across the models. This speech data consisted of audio files from only one adult female speaker, then the WER for each model was documented and compared with each other as seen in Table 1.

3.1 Google Web Speech API

Testing of the current FilBis system was done by manually feeding the speech data into the website's speech recognition system. The use of virtual audio cables ensured that the quality of the test audio was consistent throughout different researchers' testing sessions. The computed WER for the baseline model ended up at approximately 6.73%. FilBis had trouble consistently recognizing certain words such as *siyam*, *noong*, and *opo*.

3.2 Time Delay Neural Networks

Using the WER formula and the *jiwer* library, the result is 35.96% WER on roughly 1300 segmented audio segments. Upon analyzing the incorrect predictions, no common words or errors were found. The actual word versus the predicted word also did not have a predictable pattern. The probable cause for this is the lack of training data, as having only 10000 segmented audio segments with each being roughly 3 seconds long, is not enough for the model to fully capture the acoustic features of each label (see Table 1).

Table 1. WER sample for the TDNN model

Label	Predicted	WER
hindi pa po	minuto	1
nagkakapikunan sa laro	napapalo po ako	1
minsan lang po	minsan po	0.333
tinatamad po	wala po	0.5
ano po yung marka	buong katawan	1
bumaba po	wala po	0.5
dati pa	kanina	1
pag may pangarap po ako minsan po mahirap abutin	minsan lang po	0.889
si mama nakikita ang sakit namin	antibiotics	1
minsan naiinitan ako	nung isang linggo	1
minsan nalalamigan ako	eleven pm	1
hindi pa kasi ako inaantok	hindi ko po maintindihan	1
minsan lang po	tatlo	1
pag umiinom ng kape	naglalagay ng vicks	0.75

3.3 Hybrid CTC-Attention Model

With the WER formula, the model had a result of over 100% WER. Given the formula of the WER, this meant that overall, there were more errors than words in the transcriptions. Again, given all the adjustments in the hyperparameters and the consistency in the performance, the cause can be pointed towards the lack of training data for the model to consistently learn patterns from. Hybrid CTC-Attention is known to require large amounts of data in order to be efficient, which was not able to be provided in this instance.

3.4 Fine-tuned XLSR-300m Model

After fine-tuning the XLSR-300 model, The final model was evaluated to have a WER of 20.00%. With the incorrect predictions, the model seems to be problematic in differentiating its *b's* and *p's*. There were consistent errors with the model's prediction of

limampu as *limang po*. In addition, many words were either missing a character or had an extra one. Such examples of this were *po* as *pok* and *isang* as *sang*. Other errors could be found within medicinal products such as *biogesic* as *byogesic, cetirizine* as *hitrsin*, and others which may be attributed to a lack of training data.

3.5 Fine-tuned OpenAI Whisper Model

Upon fine-tuning the Whisper model, the checkpoint that had the lowest WER achieved a result of 2.61% WER. Upon further analysis of the list of incorrect predictions, some common incorrect predictions (similar to the previous model) were found to be: *limampu* and *limang po* and *nakaraang* as *nakarang* - making up roughly 50% of the errors on the testing. Further training targetting the words the model has trouble identifying would increase its performance by a considerable margin; this ability to adjust and compensate for specific weak points in its recognition is the Whisper model's main advantage over FilBis' current implementation.

Table 2. WER across all models

	Google Webspeech kit	TDNN	Base Whisper Model	Fine-tuned XLSR-300M	Fine-tuned Whisper Model
WER%	6.73	35.96	28.43	20.00	2.61

3.5 Testing With Children's Speech

As observed in Table 2, only the Fine-tuned Whisper Model had a WER comparable to the baseline, with the other 3 models having a significantly higher error rate. With only the fine-tuned Whisper model, further testing was conducted with the actual children's speech dataset to determine its feasibility.

Determining the fine-tuned Whisper model's performance involved calculating the WER for each of the 5 speakers from the schoolchildren speech dataset. Three different rounds of testing were conducted: the first round involved training the model with only children's speech data, the second round trained the model with only adult speech data, and the third round involved training the model with a mix of audio segments from both datasets. All three rounds of testing were conducted using the model by performing k-fold validations for both the adult's and children's test data sets. A split of 90%-10% was used for training and testing, respectively.

All three rounds of training with children's speech used as the test set showed a significant decline in the performance of the Whisper model. While Google's Web Speech API also struggled, with an average WER of 23.17%, the Whisper model performed even worse, yielding an average WER of 38.12%. This shows that the model faces difficulty when only given a small amount of data to be trained with. To verify this, more adult speech data beyond the single, initial speaker was used to train and test both models. The Whisper model outperformed the baseline considerably, with a WER of 4.15% compared to the Web Speech API's 9.88%.

Results show that the fine-tuned Whisper model currently underperforms compared to FilBis's ASR system (Web Speech API) when recognizing children's speech, primarily due to limited training data. However, validation using adult speech data demonstrated that with sufficient data, Whisper can surpass the Web Speech API in terms of WER. Unlike the black-box nature of the Web Speech API, Whisper offers greater long-term potential by allowing domain-specific customization—an advantage for FilBis as a children's healthcare chatbot.

The model can be improved primarily by collecting more children's speech data—ideally around 9,700 segments (about 7 hours), matching the amount used for adult training. Currently, only 570 segments (24 minutes) were used, yet the model still performed within 0.4x of the Web Speech API, showing strong potential despite limited data.

3.5 Integration into FilBis

While the model in its current state does not outperform the baseline ASR system integrated in FilBis, the Whisper model theoretically does better when supplied with more training data. The process for implementing the Whisper model into FilBis was still carried out for the purpose of future studies.

The fine-tuned Whisper model was integrated into FilBis using the small Whisper model trained on adult speech, which yielded the lowest WER in prior testing. Integration involved creating two components: a custom React hook (*useWhisperRecorder*) for client-side microphone input, and a Flask backend (*app.py*) for transcription. The *useWhisperRecorder* captures audio in the browser and sends it to the backend via an API call, where the audio is converted to the appropriate format using FFmpeg and transcribed using the Whisper model. This setup replaces the previous Web Speech Kit implementation by routing speech-to-text functionality through Whisper. Initial testing via microphone input within the chatbot showed comparable performance to the original system.

4. CONCLUSION AND FUTURE PLANS

This study explored the development and evaluation of an Automatic Speech Recognition (ASR) system for Filipino, with a focus on its performance in recognizing children's speech for a healthcare chatbot application. Several models were tested, including traditional architectures like TDNN, advanced models like Meta's XLS-R 300M and the Hybrid CTC-Attention model, and two production-grade systems: Google's Web Speech API and OpenAI's Whisper model.

Initial testing with adult speech data showed that a fine-tuned version of OpenAI's Whisper model performed the best, with an error rate of 2.61%. However, further testing with actual children's speech data showed that the improved Whisper model's performance declined significantly, garnering a 33.21% WER. This fails to surpass the current Web Speech API ASR integrated into FilBis, which presented a WER of 23.17% when tested with the same data.

Continued experimentation with the Whisper model proved its dependence on an adequate amount of training data. When trained with sufficient adult speech data, it outperformed FilBis' ASR system with the former's 4.16% to the latter's 9.89%.

With the current amount of children's speech data, the Whisper model is unlikely to outperform the baseline ASR system currently used by FilBis. However, testing shows its potential as a better long-term solution given its adaptability. Future work should focus on expanding available data for Filipino children's speech, as results suggest that Whisper could overtake existing models in Filipino when supplied with a dataset of comparable size to the adult speech dataset.

5. REFERENCES

Ang, F. (2025, March 28). FlipStar: A survey of the current state of Filipino ASR. flipvox.ph. https://flipvox.ph/post/current-state-filipino-asr/

Gandhi, Sachit. (2022). Fine-Tune Whisper For Multilingual ASR with HuggingFace Transformers. Retrieved from https://huggingface.co/blog/fine-tune-whisper

Baevski, A., Zhou, H., Mohamed, A., Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. Retrieved from https://arxiv.org/abs/2006.11477

AI at Meta. (n.d.). facebook/wav2vec2-xls-r-300m. Hugging Face. Retrieved from https://huggingface.co/facebook/wav2vec2-xls-r-300m

Radford, A., Kim, J., Xu, T., Brockman, G., Mcleavey, C., Sutskever, I. (2022). Robust Speech Recognition via

Large-Scale Weak Supervision.
https://arxiv.org/pdf/2212.04356

MozDevNet. (n.d.-a). Using the web speech API - web apis: MDN. MDN Web Docs.
https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API

Montalan, J. R., Layacan, J. P., Africa, D. D., Flores, R. I., Lopez II, M. T., Magsajo, T. D., Cayabyab, A., & Tjhi, W. C. (2025, February 19). Batayan: A Filipino NLP benchmark for evaluating large language models. arXiv.org. https://arxiv.org/abs/2502.14911

Pascual R., Ing J.A., Azcarraga J. (2023). TDNN-HMM ASR Systems on Under-Resourced Local Languages Towards Application in a Healthcare Chatbot. ICACSE, Manila, Philippines.
https://easychair.org/cfp/ICACSE2023

Shivakumar, P. G., & Georgiou, P. (2020). {Transfer Learning from Adult to Children for Speech Recognition: Evaluation, Analysis and Recommendations. Computer speech & language, 63, 101077.
https://doi.org/10.1016/j.csl.2020.101077