Machine Learning Based Network Implementation to Translate Spoken Audio to Words in TextFile

Nickmoon Mware
Department of Computer Science and Engineering
University of Nebraska-Lincol
Lincoln, Nebraska
mmware2@huskers.unl.edu

***Abstract*— Context: Automated speech recognition (ASR) systems have become common with the rise of virtual voice assistants such as Alexa. These ASR systems use machine learning algorithms to automatically transcribe the voice input to closed caption of words.**

Goal: The main aim of this work was to train a machine learning model that would be used to generate an audio input and output words as text file. This is a part of the ASR technology. Further, our unattained goal was comparing WER to determine the accuracy of developed model to existing models in order to improve ASR transcription.

Summary: Given a real voice(sound), the trained model will be used to extract the sound and output the translation to a text file as spoken. The algorithm created will be able to do optimization. If the whole statement or words are not extracted, our goal will be to extract the key words. From this work, the model we created was able to transcribe input audio file with 77% accuracy. Compared to existing models, the word error rate (WER) obtained in this work was low. Other works have shown WER of greater than 90%. Future work can further be performed to further improve this model, such as more epochs for training, using MFCC or increasing the run time during training.

Keywords*— *Automatic Speech Recognition (ASR), Speech-to-Text, Word Error Rate (WER)

I. INTRODUCTION

Automatic speech recognition. (ASR) has become ubiquitous in today's world. The ASR systems convert a speech signal input into words which can be used for text-based communication or for device controlling. Application of ASR system is widespread in several areas including virtual assistance built into mobile devices such as Google Assistant and Apple Siri, home appliances such as Alexa and in-built car systems. Speech-to-text functionality has also been implemented in automated closed captioning is a recent application of ASR systems for social media among the new generation. However, the existing ASR systems have key challenges such as poor voice recognition leading to incorrect transcription of spoken words or voice.

Advancements in machine learning and natural language processing have led to improved accuracy of the ASR system. Previous works in speech-to-text conversion lack clear metrics of comparison. Word error rate (WER) is a common performance indicator that is used to determine the accuracy of a speech recognition system. Word error rates in ASR system can occur due to background noise, complexity of speech and speaker-specific characteristics such as accents [1]. The WER of an ASR system is calculated by transcribing the speech and comparing the results. The WER is obtained using the following formula:

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{H+S+D} \quad (1)$$

Where S is word substitutions, D is word deletions and I is word insertions between the machine transcription and input words, H is the total number of hits, and N is the total number of input words. The larger the WER, the higher the difference between two transcriptions and therefore worse ARS performance. This measures the percentage of incorrect words based on the number of words processed.

The main objective herein is to develop a trained model and determine its accuracy by determining the WER of the transcribed voice input. Further, this work can be advanced by comparing the WER of the proposed model to the existing models that are open source such as Google Assistant.

II. DATA SETS

First, we'll briefly describe the data resources we used to obtain data which was trained to generate the model used for WER analysis. The resource used is freely available from open-source site with free license under Creative Commons copyright.

A. *Wikimedia Common*

The audio files used in this work were downloaded from Wikimedia common. Wikimedia common is a free open-source media repository with multiple audio formats. Audio files with wave (.wav) format were downloaded to be used as input files.

For experimental purposes and to test our model, we started the project with one audio file. The audio clip downloaded has a duration of 3 minutes and 15 seconds, with variation of pauses. The pauses between words are approximately 10 seconds long. The audio channels are mono with sample rate of 4.8 kHz, and 16 bits per sample and totaling to a size of 18,715,724 bytes (18.7 MB on disk). This file was used as audio input for further data processing.

III. AUDIO PREPARATION AND AUGMENTATION

Audio data analysis is commonly presented in two domains, its frequency or time. The audio input "Audio_Dave_thomas.wav" is from Wikimedia, a publicly accessible audio clip in the cloud. The frequency graph for the raw audio visualization is as shown in Figure 1.

We used the python librosa library for audio data processing in deep learning. Librosa is a python package for audio and music signal processing whose default is to convert the sampling rate of audio data to 22.05kHz, average the sample values from left to right and convert stereo to mono [2]. Librosa breaks the audio file to raw data processable for machine. We plot the raw data to view the demographic of the audio, spaces and ranges in frequency levels. We apply librosa

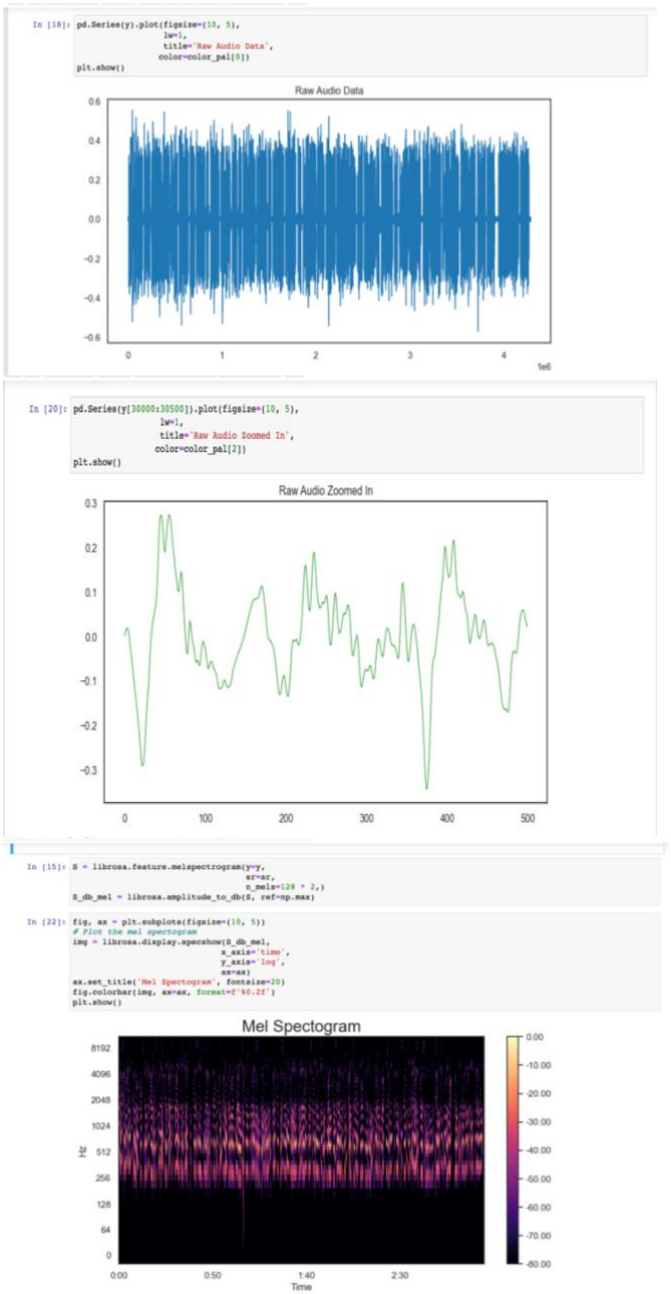

Fig. 1. *First: Raw data visualization showing 4.8kHZ audio frequency Second: Trimmed raw data and Third: Correcponding Ml Spectogram for the Raw Data after loading data into Librosa, using 128 Mel bands.*

effects, mainly color and trimming to produce spectrograms for the audio. The feature analyses conducted by Librosa produces two-dimensional (2D) arrays output which is stored as numpy.ndarray [2]. During this process, the output from Librosa is imported into 2D NumPy arrays using imported python numpy library. Numpy generally provides multidimensional array object and various routines for fast operation on arrays.

Mel frequency scale allow humans to perceive sound frequency and is used to represent audio signal [3]. Librosa provides a platform for both Mel-scale spectrogram and Mel-Frequenct Cepstral Coefficients (MFCC). Mel-scale spectrogram were generated in Librosa as shown in Figure 1.

To have exact matching, we used the English dictionary as the target labels for characters from the transcription using the Python Alphabet library. Character ID's were constructed from vocabularies built in the transcript. Once the audio files in the WAV format were preprocessed, the data was used to build and train a model to recognize the words in the input audio. The WAV audio file was cut into short segments, mel spectrograms were generated and this was used as input for training the model. Other python libraries, NumPy and Pytorch, were also used in this process. The validation set was split into two halves in order to obtain a test set and a validation set of data. Since the data was transformed into spectrogram images, the convolutional

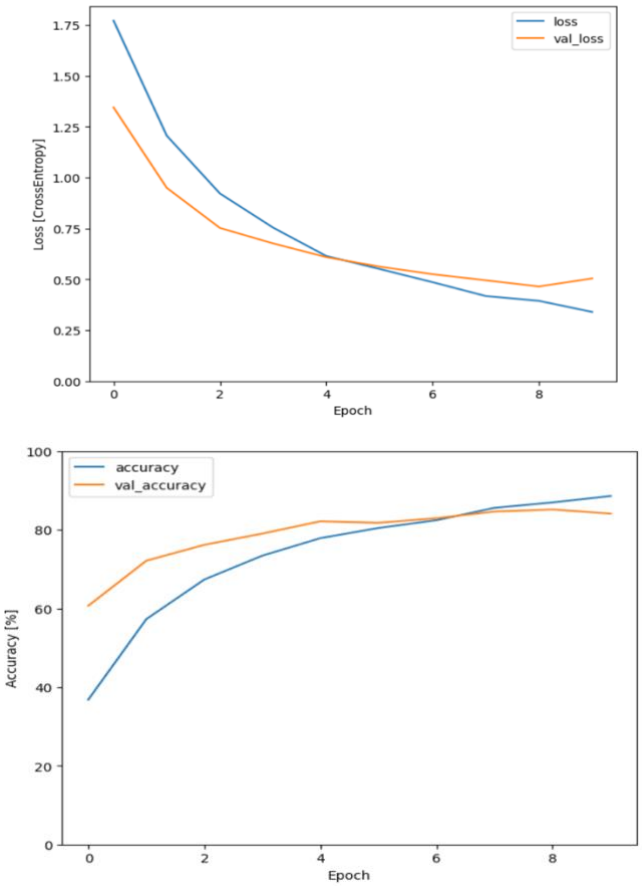

Fig. 2. Loss Curve and Accuracy Curves from Model Training and Validation

neural network (CNN) was used to generate the model. Aggregate statistics for mean and standard deviation were computed in the training data using the normalization layer. The model was trained using 10 epochs and training and validation loss curves were obtained to check model's improvement during the course of the training (Figure 2). Once the model was trained, the entire audio file was used as input data to test the models functionality. WER was further calculated to determine the effectiveness and accuracy of the generated model.

When building the training model, we had key consideration in parameters to make, our selection included: - 10 epochs to train the data, our batch size was 7 as we split our audio file to seven chunks, training data quantity~ we had one data taken out of the source.

IV. RESULTS

Prior to model training, the initial ASR model was used to generate a .txt file of the input audio (Figure 3). The accuracy of the data output was 27.3%, with very low accuracy of spoken word. After model training, the accuracy level 77.3% accuracy (Figure 4a). This accuracy rate was higher compared to the first trial. As observed in the accuracy curves, with increase in epoch, the accuracy rate of the trained model significantly increased (Figure 2). More epochs will probably be needed to further improve the accuracy of this proposed model. Further, a more efficient code can be written to generate better results. Although this is a lower number than what was anticipated, it is a steppingstone and encouraging since our model works. We aim to improve our code model and run it again with changes in duration. Compared to current state of art, our accuracy levels are low, as shown in table 1 [9], [10], [11], [12],[13].

TABLE I. STATE OF ART ACCURACY OF ASR MODEL

Author	**Table Column Head**	
	Main work	***WER (%)***
Chen et al	HTS-AT – Tramsformer model with hierarchial structure and token-semantic modules	97%
Elizalde et al	CLAP – CNN model pretrained by natural language supervision	96.7%
Gong et al	AST – Pure attention model pretrained on AudioSet	95.7%
Huang et al	AclNet – CNN with mixup and data augmentation	85.65%
Aytar et al	Soundnet – 8-layer CNN with transfer learning from unlabeled videos	74.2%

V. DISCUSSION

Our model was able to successfully transcribe the audio file to text with an accuracy of 77%. We started off with a very poor transcription of about ~27% accuracy before applying Connection Temporal Classification (CTC) – CTC is a type of neural network output and associated scoring function, for training recurrent neural networks such as LSTM networks to tackle sequence problems where the timing is variable. Our audio file is large which lead to a poor first translation, we thus split our audio file to chunks, we also created 0.5 seconds of silence chunks to cater for low frequencies.

We build a recognizer function in python which imported our audio file and generates a spectrogram. The function is useful as it listens to the audio and builds response object, these response objects are necessary to inform us whether audio processing is a success, error, or the transcription is ongoing. Split audio into chunks – we ran our code and noticed that our transcription was a success, but the output still generated words, we checked our model and epoch size but noticed that was not the error but the length of our audio file which couldn't be fully read. We hence split the audio file in the recognizer function to seven chunks and implemented a loop which ensured each chunk was fully transcribed before moving on to the next audio chunk. After running our model, we extract the words from the transcription to a text file in the same path as the source code.

Our most challenging task was during code debugging and configuring our training data. As initially mentioned, when the model first ran, it produced a very low WER and as such many changes had to be made to the code to make sure we get better results. We had to do more research and try various approaches to our model to make it work better. At the time of debugging, our model would shut down meaning our model needed further improvement. The WER is still high, and this remains to be the highest limitation facing our paper, we aim to improve our model and thus gain better results in terms of classification method and ASR system accuracy.

Our model currently analyzes only one audio file, other research related to speech-to-text conversion usually use a large dataset (metadata) of audio files, as demonstrated by de Pinto et al., [7]. An area for improvement would be increasing our data size and including more audios for training. Also, our epoch value for our training model is low and improving it might lead to better results in the transcription. Lastly, our training model only runs for ~6 hours, which is factual considering the size of our data but running our model for longer periods might yield better results.

Mel Spectrograms was used as a feature to train our model, instead of using Mel Spectrograms, we could use Mel Frequency Cepstral Coefficients (MFCC) which is the state of art in human sounds formalization. MFCC is a python library that can be used to extract only the most essential frequency coefficients. Various research, including Guisepe et al. who achieved an accuracy score of .95 when using MFCC to test emotion detection from spoken language, is a great incentive to try MFCC in our model with the aim of getting better accuracy scores.

Research by Han et al introduced ContextNet which an encoder that analyzes global context information by squeezing and excitation of modules. We are currently studying their approach while looking up other modeling ideas to improve our accuracy. Thus far, we have worked towards implementing some suggestions we received from friends to improve our model such as increasing the size of input, increasing the number of epochs and batch number which is gradually showing improvement [4]. Another approach to improve WER is the denoising approach, the idea was introduced by Kinoshita et al [5] where they demonstrated a single-channel time-domain denouncing approach, which provided 30% WER for ASR's, as shown in their study.

VI. FUTURE WORK

Further work can be developed to further illustrate the effects of speaker variability on the accuracy and performance of these speech recognition tools. Speaker variability occurs naturally mainly due to differences in biological characteristics as influenced by gender, rate of speech, tonal variation, accents, vocal effort and speaking style. Recent inventions in speech recognition have been geared towards accent identification, whereby the speech recognition system selects an appropriate language base to transcribe the spoken language based on users' accent.

Hannah et al. in their research asked the question, "To what extent does ASR accuracy vary across language background?" and their answer proved that student language background has a significant effect on ASR accuracy when evaluating the use of ASR for young language learners [6]. This introduces a new research domain under our work, language barrier, our long-term goal is to improve our model and enable to capture different accents which is a challenge facing current ASR's.

Allison et al also present racial disparities in ASR's, in their research conducted, they show that black speakers had a WER of 0.35 while white speakers had a WER of 0.19 in an experiment to evaluate the 5 state-of-art ASRs developed by Apple, Amazon, Google, Microsoft and IBM. This ties to our end-goal in improving accent recognition for various English-speaking nationalities with the aim of reducing WER for common ASR's [8].

VII. CONCLUSION

We have used CTC for our ML model to output spoken audio to text, our result shows this technique is among the best in the field of NLP and Speech to Text research. We observed an accuracy of 0.7735 in our experiment, current models have a higher accuracy and thus room for improvement on our current model. Our research is expected to aid future studies on natural language processing and implementation of ASR's, we plan to come up with a better model for audio translation and address the bias in different accent recognition by ASR's.

In summary, this work answers the following questions to add to the state-of-art of ASR:

1. What is the key problem?

   The key problem is the inability of Automated Speech Recognition (ASR) tool to fully capture and transcribe spoken audio. Speech-to-Text functionality of various ASR's is unexplainable and lacks a clear measurement metric.

2. Why is it essential?

   Use of ASRs such as Apple Siri is on the rise and having a model which fully understands user commands is good to have and increases users' confidence in using the voice recognition systems/ automated speech recognition tools.

3. What are the previous works, including advantages and disadvantages?
   Previous work includes [4], the authors develop a model to classify emotions from spoken language, this kind of research conducted often develop a classifier to understand various sound patterns and group them according to their differences. Our project is different in that we do not want to classify spoken audio but instead produce a transcription which will help the readers understand the process of speech-to-text conversion in deep learning. Current ASR tools are not open source and users do not know what happens behind the scenes or how the translation is processed by the machine.
4. What are your solutions, and why would they be better than the previous ones?
   We develop a deep learning model and train it to capture key words and statements from an audio file and in return produce a text file with the transcription and the accuracy of the transcription. Our model is a step ahead as it also provides the accuracy of the transcription, one key challenge facing ASRs is bias WERs and lack of metric analysis.
5. How would it be achievable? (e.g., required steps)
   Our goal was achievable by creating a model in deep learning that took in audio in wave file, converted the file into spectrograms which is machine readable and used CTC to decode what was said into characters which are outputted in a textfile. All this code was written in python.

Our research is expected to aid future studies on natural language processing and implementation of ASR's, we plan to come up with a better model for audio translation and address the bias in different accent recognition by ASR's.

ACKNOWLEDGMENT

I would like to acknowledge my professor for the support and academic guidance throughout the semester of Fall 2022. My teammates and classmates helped me brainstorm on this idea and polish it and for that I am grateful. I would also like to thank Didier Ishimwe who helped me crosscheck my code and debug.

REFERENCES

[1] Meyer, B. T., Mallidi, S. H., Kayser, H., & Hermansky, H. (2017, March). "Predicting error rates for unknown data in automatic speech recognition". In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5330-5334). IEEE.

[2] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (Vol. 8, pp. 18-25).

[3] Stevens, S. S., Volkmann, J., & Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, *8*(3), 185-190.

[4] Han, Wei, et al. "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context." arXiv preprint arXiv:2005.03191 (2020).

[5] Kinoshita, Keisuke, et al. "Improving noise robust automatic speech recognition with single-channel time-domain enhancement network." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

[6] Hannah, L., H. Kim, and E. E. Jang. "Investigating the effects of task type and linguistic background on accuracy in automated speech recognition systems: Implications for use in language assessment of young learners." Language Assessment Quarterly (2022): 1-25.

[7] de Pinto, Marco Giuseppe, et al. "Emotions understanding model from spoken language using deep neural networks and mel-frequency cepstral coefficients." 2020 IEEE conference on evolving and adaptive intelligent systems (EAIS). IEEE, 2020.

[8] Koenecke, Allison, et al. "Racial disparities in automated speech recognition." Proceedings of the National Academy of Sciences 117.14 (2020): 7684-7689.

[9] Chen, Ke, et al. "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection." *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[10] Gong, Yuan, Yu-An Chung, and James Glass. "Ast: Audio spectrogram transformer." *arXiv preprint arXiv:2104.01778* (2021).

[11] Elizalde, Benjamin, et al. "Clap: Learning audio concepts from natural language supervision." *arXiv preprint arXiv:2206.04769* (2022).

[12] Huang, Jonathan J., and Juan Jose Alvarado Leanos. "Aclnet: efficient end-to-end audio classification cnn." *arXiv preprint arXiv:1811.06669* (2018).

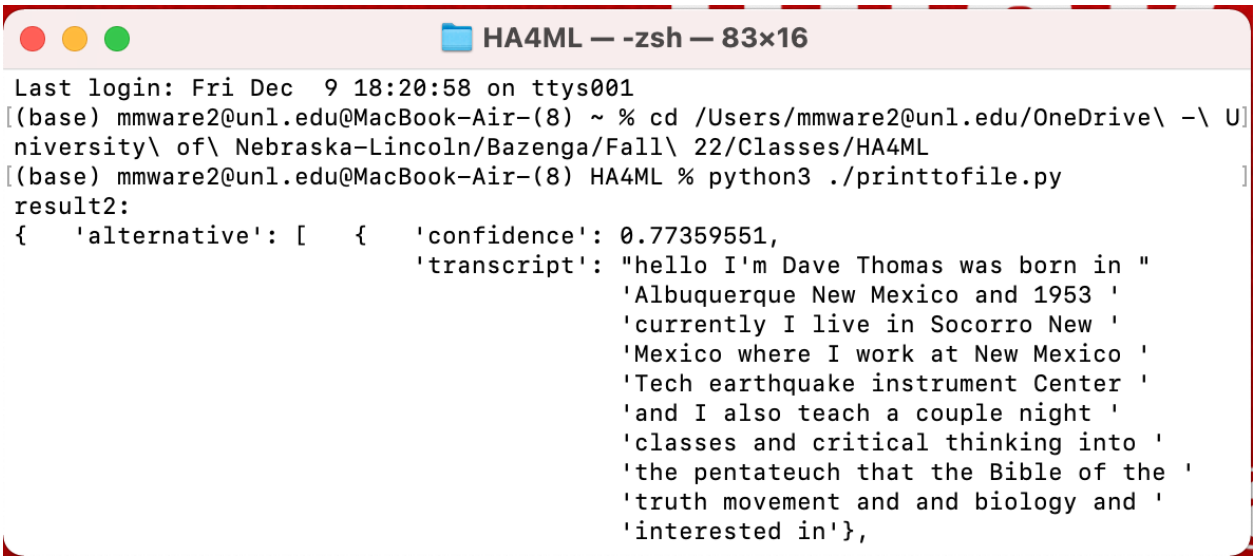

Figure 3. Accuracy Level of Model Input

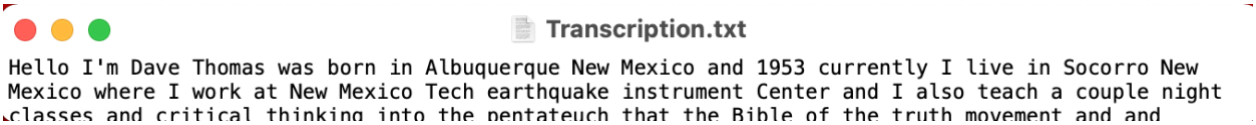

Figure 4. Excerpt of Text File Ouput

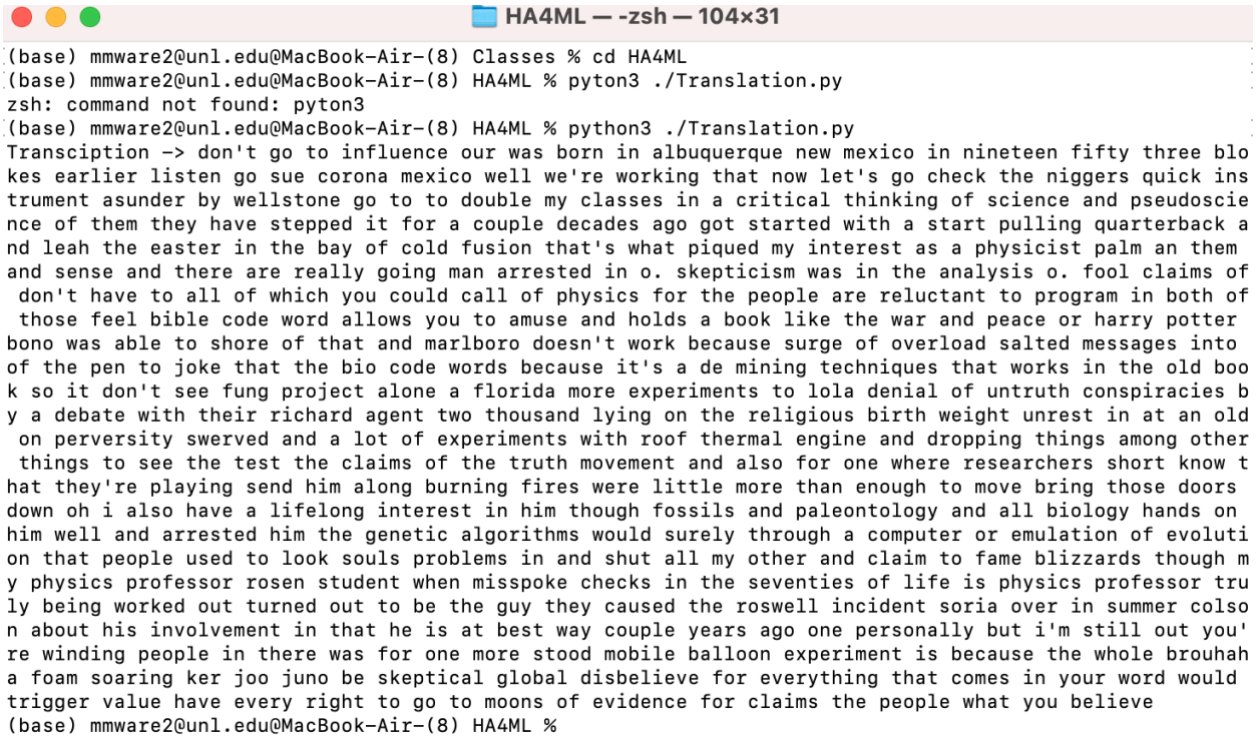

Figure 5. Transcription of Audio-to-Text Model with Poor Transcription Rate