HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Project 1 report

Speech-to-Text System for Vietnamese and English using Deep Learning

---

Submitted by

Vu Tuan Truong - 20225535

Guided by

Prof. Pham Dang Hai

Hanoi, January 2025

Abstract

This project presents a speech-to-text (STT) Windows app designed to transcribe Vietnamese and English audio, with a particular focus on accurate transcription of lecture recordings. The app leverages the power of deep learning by incorporating a diverse range of state-of-the-art models and APIs. To enhance the system's robustness, a Voice Activity Detection (VAD) module is used to preprocess audio by removing silent segments, potentially improving transcription accuracy. Performance evaluation is performed using the word error rate (WER) and character error rate (CER) on a custom dataset, demonstrating the effectiveness of the chosen models and the impact of preprocessing techniques. This project highlights the strengths and weaknesses of various deep learning approaches to speech-to-text in both Vietnamese and English. The developed system, along with an evaluation script and an installer, is publicly available for further research and development.

Contents

List of Tables

Part I: Introduction

1 Project Overview

This project focuses on the development of a speech-to-text (STT) system capable of transcribing Vietnamese and English audio. The system is designed to be a practical and user-friendly tool, particularly useful for generating accurate transcriptions of lecture recordings. Our approach integrates a diverse set of deep learning models and APIs, namely OpenAI's Whisper and Facebook's Wav2Vec 2.0 for the local models and DeepGram and AssemblyAI for the API models.

The app is controlled through a graphical user interface (GUI) developed using the PyQt6 framework. Using the GUI, the user can easily select models, languages, audio files and tweak various settings, including the API key configuration and VAD toggles. The source code and evaluation script are made publicly available to facilitate further use and development. The project also includes an installer for a smooth setup on Windows computers.

For the evaluation of the various models used in the project, WER and CER are used to rigorously evaluate the performance of each model. Since this project's focus is on lecture recordings, especially Vietnamese ones, we put together a unique dataset of news and lecture audio for evaluating the models. Furthermore, we provide a standalone evaluation script that allows anyone to test and evaluate the available models and APIs on their own custom datasets.

2 Motivation

With the increasing availability of audio and video content nowaday, the need for effecient STT systems keeps growing more and more, especially in educational settings with the spreading of online learning. Therefore, automatic transcription of lectures, meetings and other type of recoordings can significantly improve accessibility for everyone. However, building robust STT systems presents numerous challenges, especially for languages like Vietnamese, including diverse accents, background noise and unclear or fast pronunciation or words.

This project is motivated by the desire to try to address these challenges and develop a practical STT app that allows for the transcription of real-world audio. We put cutting-edge models to the test on Vietnamese audio of varying quality, to see how they perform and whether more research and training could unlock even better results.

Part II: Background and Related Work

3 Speech Recognition Concepts

Speech recognition, also known as automatic speech recognition (ASR) or speech-to-text (STT), is the ability of a computer system to identify and convert spoken language into written text. It is a multidisciplinary field that draws upon concepts from linguistics, computer science, and electrical engineering. The fundamental goal of an STT system is to accurately transcribe the spoken words, regardless of variations in pronunciation, accent, speaking rate, or background noise.

A typical STT system consists of several key components:

- **Acoustic Modeling:** This component analyzes the raw audio signal and extracts relevant acoustic features that represent the sounds of speech. These features often include spectral information, such as Mel-Frequency Cepstral Coefficients (MFCCs) or filter bank energies, which capture the energy distribution across different frequency bands. The acoustic model is typically trained on a large dataset of speech recordings and their corresponding text transcripts. It learns to map sequences of acoustic features to phonemes (the basic units of sound in a language) or other subword units.
- **Language Modeling:** This component provides context and helps the system predict the most likely sequence of words given the acoustic input. Language models are statistical models that capture the probabilities of word sequences in a particular language. They are trained on vast amounts of text data and learn the grammatical structure and vocabulary of the language. N-gram models and, more recently, neural network-based language models are commonly used.
- **Decoding/Search:** This component combines the information from the acoustic, language, and pronunciation models to find the most likely sequence of words that corresponds to the input audio. Efficient search algorithms, such as the Viterbi algorithm or beam search, are used to explore the vast space of possible word sequences and identify the best hypothesis.

Historically, STT systems relied on Hidden Markov Models (HMMs) combined with Gaussian Mixture Models (GMMs) for acoustic modeling and N-gram models for language modeling. However, in recent years, deep learning has revolutionized the field, leading to significant improvements in accuracy and robustness. End-to-end deep learning models, which directly map audio input to text output, have become increasingly popular, simplifying the pipeline and allowing for joint optimization of all components.

4 Challenges in Speech Recognition

Despite significant advancements in recent years, speech recognition remains a challenging task due to the inherent variability and complexity of spoken language. Some of the key challenges include:

- **Speaker Variability:** Individuals have different accents, speaking rates, vocal tract characteristics, and pronunciation patterns. An STT system must be robust to these variations to accurately transcribe speech from diverse speakers.
- **Acoustic Environment:** Background noise, reverberation, and other acoustic distortions can significantly degrade the quality of the audio signal and make it difficult to distinguish speech sounds.
- **Speaking Style:** Speech can vary from formal and clear (e.g., news broadcasts) to informal and conversational (e.g., casual conversations, lectures). Spontaneous speech often contains disfluencies, such as hesitations, repetitions, and corrections, which can be challenging for STT systems to handle.
- **Vocabulary and Language:** Out-of-vocabulary (OOV) words, rare words, and domain-specific terminology can pose difficulties for STT systems, particularly if they have not been encountered during training. The complexities of grammar, syntax, and semantics also add to the challenge.
- **Code-Switching and Multilingualism:** In many regions, speakers may switch between multiple languages within a single conversation or even within a sentence. This code-switching phenomenon poses a challenge for STT systems that are typically trained on a single language.
- **Ambiguity:** Spoken language can be inherently ambiguous. Homophones (words that sound the same but have different meanings) and the lack of clear word boundaries in continuous speech can make it difficult to determine the correct transcription.
- **Computational Resources:** State-of-the-art deep learning models for STT can be computationally expensive to train and deploy, requiring significant processing power and memory, particularly for real-time applications.
- **Data Scarcity**: For some languages and domains, there is a lack of sufficient transcribed speech data to train accurate and robust models.

These challenges, especially when combined, can significantly impact the accuracy and reliability of STT systems. Researchers continue to develop new techniques to address these issues and improve the robustness of speech recognition technology.

5 Deep Learning for STT

Deep learning has revolutionized the field of speech recognition, leading to significant improvements in accuracy and robustness. Unlike traditional HMM-GMM-based systems, deep learning models can learn complex patterns and representations directly from raw data, reducing the need for manual feature engineering.

Several deep learning architectures have been successfully applied to STT:

- **Convolutional Neural Networks (CNNs):** CNNs have been used to model local patterns in the spectral representation of speech signals, similar to how they are used for image recognition.
- **Recurrent Neural Networks (RNNs):** RNNs, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, are well-suited for processing sequential data like speech. They can capture temporal dependencies and context information.
- **Transformer Networks:** The Transformer architecture, based on the self-attention mechanism, has shown remarkable performance in various NLP tasks, including STT. Models like the Transformer and Conformer have achieved state-of-the-art results on many speech recognition benchmarks.
- **End-to-End Models:** End-to-end models, such as Connectionist Temporal Classification (CTC) and sequence-to-sequence models with attention, directly map speech input to text output, simplifying the traditional STT pipeline.

Recent research has focused on areas such as:

- **Unsupervised and Semi-Supervised Learning:** To reduce the reliance on large amounts of labeled training data, researchers are exploring unsupervised and semi-supervised learning methods that can leverage unlabeled or partially labeled data.
- **Transfer Learning and Multilingual Models:** Techniques for transferring knowledge learned from one language to another are being developed to improve performance on low-resource languages.

6 Focus on Whisper, Wav2Vec 2.0, Deepgram, and AssemblyAI

This project utilizes and evaluates several state-of-the-art models and APIs for speech-to-text:

6.1 OpenAI Whisper

Whisper [4] is a family of Transformer-based, multilingual, and multitask models developed by OpenAI. It is trained on a massive dataset of 680,000 hours of multilingual and multitask supervised data collected from the web.

Key Features:

- **Architecture:** Whisper models use an encoder-decoder Transformer architecture. The encoder processes the input audio, represented as a log-Mel spectrogram, and the decoder generates the corresponding text.

- **Multilingual:** Whisper models are trained on data from 98 different languages, enabling them to perform speech recognition and translation for a wide range of languages.
- **Multitask:** The models are trained on various tasks, including speech recognition, speech translation, voice activity detection, and language identification.
- **Model Sizes:** Whisper models come in different sizes, ranging from ”tiny” (39M parameters) to ”large” (1550M parameters) and recently ”turbo” (809M parameters), which is a distilled version of the ”large” model, offering a trade-off between accuracy and computational cost.
- **Performance:** Whisper has demonstrated state-of-the-art performance on various benchmarks, particularly in zero-shot settings (i.e., without fine-tuning on the target task or domain).

In this project, we utilize the ‘faster-whisper‘ implementation, which provides an optimized version of the Whisper models using CTranslate2, a fast inference engine for Transformer models. We evaluate different sizes of the Whisper models (tiny, base, medium, large and turbo) to analyze the impact of model size on accuracy and speed.

6.2 Facebook Wav2Vec 2.0

Wav2Vec 2.0 [1] is a self-supervised framework for learning speech representations from raw audio data. It is developed by Facebook AI Research (FAIR).
Key Features:

- **Self-Supervised Pre-training:** Wav2Vec 2.0 is pre-trained on a large amount of unlabeled speech data using a contrastive learning objective. The model learns to distinguish between true speech segments and distractors in a quantized latent space.
- **Architecture:** The model consists of a multi-layer convolutional feature encoder that takes raw audio as input and produces a sequence of latent speech representations. A Transformer architecture then processes these representations to capture contextual information.
- **Fine-tuning:** After pre-training, the model can be fine-tuned for specific downstream tasks, such as speech recognition, using a relatively small amount of labeled data.
- **Performance:** Wav2Vec 2.0 has achieved state-of-the-art results on various speech recognition benchmarks, particularly in low-resource settings.

In this project, we leverage the ‘transformers‘ library from Hugging Face to access pre-trained Wav2Vec 2.0 models. We specifically focus on a Vietnamese-language model [3] [2] fine-tuned on the VLSP 2020 ASR dataset. We also explore the integration of Silero VAD with Wav2Vec 2.0 to potentially improve its robustness.

6.3 Deepgram API

Deepgram is a cloud-based speech recognition platform that offers a powerful API for developers. It leverages deep learning models trained on a vast amount of diverse audio data.
Key Features:

- **Accuracy:** Deepgram claims to provide highly accurate transcriptions, even in noisy environments and with accented speech.
- **Speed:** Deepgram offers fast transcription speeds, both for real-time and pre-recorded audio.
- **Features:** The Deepgram API provides various features, including speaker diarization, keyword boosting, profanity filtering, and punctuation.
- **Language Support:** Deepgram supports a wide range of languages and dialects.
- **Scalability:** As a cloud-based API, Deepgram can handle a large volume of transcription requests.

In this project, we integrate the Deepgram API into our system to compare its performance with local models. We utilize the Deepgram Python SDK to send audio files to the API and receive the transcribed text.

6.4 AssemblyAI API

AssemblyAI is another cloud-based speech recognition platform that provides an API for developers. It uses advanced deep learning models to deliver accurate and efficient transcriptions.
Key Features:

- **Accuracy:** AssemblyAI claims to achieve high accuracy through its use of state-of-the-art models and training techniques.
- **Features:** The AssemblyAI API offers a variety of features, including speaker diarization, sentiment analysis, topic detection, and custom vocabulary.
- **Language Support:** AssemblyAI supports a broad range of languages and accents.
- **Real-time Transcription:** AssemblyAI provides real-time transcription capabilities, making it suitable for live captioning and other real-time applications.
- **Ease of Use:** The AssemblyAI API is designed to be developer-friendly and easy to integrate.

In this project, we integrate the AssemblyAI API to evaluate its performance and compare it with other models and APIs. We use the AssemblyAI Python SDK to interact with the API, send audio files for transcription, and receive the results.

By incorporating these diverse models and APIs, our project provides a comprehensive evaluation of the current state-of-the-art in speech-to-text technology for both Vietnamese and English, with a focus on Vietnamese audio. The comparison between local models (Whisper, Wav2Vec 2.0) and cloud-based APIs (Deepgram, AssemblyAI) highlights the trade-offs between accuracy, speed, computational resources, and ease of use, providing valuable insights for researchers and developers in the field of automatic speech recognition.

Part III: Model Evaluation

7 Evaluation Setup

7.1 Dataset

Our custom dataset consists of 7 files in total, 4 of which are audios from various VTV24 news segments and the other 3 are lecture audios, all of which are taken from Youtube. These files exhibit a range of audio qualities, including variations in noise levels, pronunciation clarity (from perfect to accented), speaking speed and mixing of English words. The varying quality in our dataset ensures that we can thoroughly evaluate the models and figure out there strengths and weaknesses in transcribing different audio conditions.

7.2 Audio Preprocessing

Since all of the models, both local and API-based, perform best on mono-channel audio with the framerate of 16 kHz, all of the audios are first converted to that format first. Following this conversion, the audio data are extracted into a numpy array representation. This format is suitable for input into the local models or be the payload for the API models.

7.3 Text Preprocessing

For the seven files in our custom dataset, the transcription ground truths are meticulously created manually to guarantee the accuracy and quality. However, our primary focus is on how well each model capture what is being spoken, rather than the quality of the transcriptions' presentation. Therefore, we preprocessed these ground truth transcriptions before using them for evaluation. This preprocessing involved removing all punctuations and special characters beside mathematical symbols, remove all newlines and redundant whitespace and then split all words, or characters based on the metric used, into an array.

7.4 Evaluation metrics

To rigorously evaluate the performance of our speech-to-text models, we decided to use two widely employed metrics in the field of speech detection: Word Error Rate (WER) and Character Error Rate (CER). By measuring errors at different granularity, these two metrics together can provide us with valuable insight on the accuracy of each model.

7.4.1 Word Error Rate (WER)

Word Error Rate (WER) is one of the most common metric for evaluating the accuracy of speech recognition systems. the metric measures the distance between the hypothesis and its ground truth by using Levenshtein distance. Levenshtein distance is measured by calculating the minimum number of modifications we need to perform to transform the hypothesis into the ground truth. The modifications are divided into three types:

- **Substitutions (S):** A word in the hypothesis is replaced by a different word in the reference.
- **Deletions (D):** A word in the reference is omitted in the hypothesis.
- **Insertions (I):** A word is added to the hypothesis that is not present in the reference.

The WER encapsulate all three types of modifications, as can be seen in its formula:

$$\text{WER} = \frac{S + D + I}{N}$$

Where:

- S is the number of substitutions.
- D is the number of deletions.
- I is the number of insertions.
- N is the total number of words in the reference text

WER typically falls between 0 and 1, so it is usually represented as a percentage. The lower the WER is, the better the transcription. A 0% WER signifies perfect transcription with no errors, while a WER greater than or equal to 100% signifies that the hypothesis and ground truth have no relation to each other.

Example:

- **Reference:** the quick brown fox jumps over the lazy dog
- **Hypothesis:** the quack brown fox jump over lazy dog

In this example:

- $S = 2$ (quick -> quack, jumps -> jump)
- $D = 1$ (the second ”the” is missing)
- $I = 0$
- $N = 9$

Therefore: WER $= (2 + 1 + 0)/9 = 3/9 = 0.333$ or 33.3%

7.4.2 Character Error Rate (CER)

Character Error Rate (CER) is similar to WER but operates at the character level instead of the word level. It measures the minimum number of character-level edits (substitutions, deletions, and insertions) needed to transform the hypothesis into the ground truth.

The CER is calculated using the same formula as WER, but with S, D, I, and N now representing character-level counts:

$$\text{CER} = \frac{S + D + I}{N}$$

Where:

- S is the number of character substitutions.
- D is the number of character deletions.
- I is the number of character insertions.
- N is the total number of characters in the reference text.

CER is also expressed as a percentage, with a lower CER indicating better performance.

Example:

- **Reference:** the quick brown fox
- **Hypothesis:** the qick brown foxex

In this case:

- $S = 0$
- $D = 2$ (foxes -> fox)
- $I = 1$ (qick -> quick)
- $N = 19$) (including spaces)

Therefore: CER $= (0 + 2 + 1)/19 = 3/19 \approx 0.158$ or 15.8%

7.4.3 Why Use Both WER and CER?

WER and CER are both valuable metrics for evaluating the accuracy of speech-to-text transcriptions. However, when applied to a language like Vietnamese, examining both metrics in conjunction provides a more comprehensive understanding of the model's performance than relying on either metric individually. The interplay between these two metrics reveals a nuanced picture of transcription accuracy, especially in light of the unique linguistic characteristics of Vietnamese.

In particular, Vietnamese is an isolating language, meaning that words are often single syllables and separated by spaces. This distinct separation of words makes WER a particularly effective metric for assessing transcription accuracy. WER directly quantifies the number of incorrectly transcribed words, giving us a clear and intuitive measure of how well the model has grasped the overall semantic content of the audio.

On the other hand, Vietnamese is a tonal language with diacritics that significantly affect meaning of the word. Furthermore, the language contains numerous homophones or near-homophones, where words with similar pronunciations also tend to have similar spellings, creating a source of confusion for transcription models. By analyzing errors at the individual character level, CER offers a more granular evaluation of the transcription, especially regarding tonal markings and subtle phonetic differences. This not only give us more details that WER generally miss but also valuable insights into the underlying reasons for the model's mistakes. For instance, it can highlight whether errors stem from misinterpreting tones, confusing similar-sounding words, or struggling with specific phonetic features.

Part IV: Results and Discussion

8 Experimental Results

This section presents the experimental results obtained from evaluating the performance of each model on our Vietnamese audio dataset. The evaluation was conducted using the script detailed in Chapter 7, and the results were logged for analysis. We assessed the performance using two key metrics as presented above: Word Error Rate (WER) and Character Error Rate (CER). The models evaluated include local models (OpenAI Whisper, Facebook Wav2Vec 2.0) and API-based services (Deepgram and AssemblyAI). The local models were tested both with and without Voice Activity Detection (VAD) to analyze its impact.

8.1 Evaluation Data

The dataset consists of:

- 3 lecture audio files (lecture1.mp3, lecture2.mp3, lecture3.mp3) with corresponding text transcriptions (lecture1.txt, lecture2.txt, lecture3.txt).

- 4 news segment audio files (vtv1.wav, vtv2.mp3, vtv3.mp3, vtv4.mp3) with corresponding text transcriptions (vtv1.txt, vtv2.txt, vtv3.txt, vtv4.txt).

8.2 Evaluation Results

The following tables summarize the evaluation results for each model, segmented by the use of VAD (where applicable) and categorized by dataset type.

Table 1: Performance of OpenAI Whisper Models

Model Size	VAD	Dataset	Avg. WER (%)	Avg. CER (%)
Medium	False	All	33.65	24.17
		VTV	17.38	9.84
		Lecture	55.36	43.27
	True	All	23.16	15.01
		VTV	18.86	11.22
		Lecture	28.88	20.07
Turbo	False	All	31.69	23.67
		VTV	34.03	28.04
		Lecture	28.57	17.84
	True	All	19.02	12.58
		VTV	18.29	12.67
		Lecture	20.00	12.46

Table 2: Performance of Facebook Wav2Vec 2.0 Model

VAD	Dataset	Avg. WER (%)	Avg. CER (%)
False	All	30.97	18.38
	VTV	25.46	14.82
	Lecture	38.32	23.13
True	All	27.68	16.54
	VTV	21.84	12.47
	Lecture	35.47	21.97

Table 3: Performance of API-Based Models

Model	Dataset	Avg. WER (%)	Avg. CER (%)
Deepgram	All	25.73	18.92
	VTV	23.15	17.44
	Lecture	29.17	20.90
AssemblyAI	All	26.03	18.17
	VTV	24.65	18.39
	Lecture	27.87	17.87

9 Discussion

9.1 Performance Analysis

The evaluation results demonstrate several key findings:

- **General evaluations:**
  - For local models without VAD, we can see that the overall WER is about 30%, which is significantly higher than what was reported in the original research papers. This suggest that while these models performs really well on popular datasets, their performance takes a noticeable hit when faced with the messy real-world data.
  - Wav2Vec2 has a lower CER compared to both Whisper Medium and Turbo, this suggests that Wav2Vec2's training approach of combining unsupervised pre-training with supervised fine-tuning helps it better handles unique phonetic and linguistic characteristics with very limited fine-tuning data. This could mean better performance when encountering unusual words, words said with heavy accents or in very noisy environment.
  - The API-based models (Deepgram and AssemblyAI) showed competitive performance compared to the local models, especially when considering the computational cost and ease of use associated with cloud-based solutions.
- **Impact of VAD:** The use of VAD generally improved the performance of local models. This suggests that removing silent segments can enhance the accuracy of these models by focusing on relevant speech portions, preventing the models from hallucinating during silent segments.
- **Model Size:** For Whisper, the "Turbo" model clearly outperformed the "Medium" model, despite having quite similar number of parameters. It is also much more stable, having pretty consistent WER and CER across different audio files. This indicates that while larger model sizes clearly lead to better accuracy, pruned model of the larger models can still has reasonably comparable results with much lower computational costs.

- **Dataset Differences:** Performance varied between the ”VTV” and ”Lecture” datasets. The models generally performed better on the ”VTV” dataset, potentially due to the more formal and clear speech characteristics of national broadcast audio compared to lecture recordings.

9.2 Discussion of Results

The results highlight the benefits of using distilled version of large models, with comparable performance at a much lower computation cost. The use of VAD demonstrates a clear benefit in improving accuracy, particularly for less formal audio like lecture recordings. The API-based models provide a viable alternative for users who have limited local computational resources.

The differences in performance between the ”VTV” and ”Lecture” datasets suggest that the models are sensitive to the characteristics of the audio. Further investigation into the specific error patterns on each dataset could reveal areas for improvement in preprocessing or model training.

Overall, the evaluation results provide valuable insights into the strengths and weaknesses of different STT models for the Vietnamese language. These findings can guide further development and optimization of the system, as well as inform the choice of model for specific applications.

It is important to note that these results are specific to the evaluated dataset, which is still small, and may vary depending on the characteristics of the audio, the quality of the transcriptions, and the specific implementation details of each model. However, the observed trends provide a useful starting point for understanding the performance landscape of these STT models and APIs.

Part V: Conclusion and Future Work

10 Conclusion

This capstone project successfully developed a user-friendly speech-to-text system for transcribing Vietnamese and English audio, leveraging a diverse range of deep learning models (OpenAI’s Whisper, Facebook’s Wav2Vec 2.0) and APIs (Deepgram, AssemblyAI). We have evaluate and compare these models in the context of Vietnamese audio, using WER and CER, which highlighted the impact of model size, VAD usage, and dataset characteristics on performance.

11 Concluding Remarks

Our findings demonstrate the significant potential of current STT technology for real-world applications, particularly in transcribing lecture recordings. While larger models like Whisper ”turbo” and API-based solutions offer high accuracy, the choice of model depends on the specific use case and available resources. This project provides a solid foundation for further advancements in STT, with future work focusing on enhancing robustness, exploring new

architectures, and expanding language support. The developed system serves as a valuable tool for improving accessibility and efficiency in various domains, paving the way for more sophisticated and versatile speech recognition applications.

12 Further Work

This project provides a solid foundation for our future development and research in speech-to-text. Several avenues can be explored to enhance the system's capabilities and expand its applicability:

- **Further Enhancement of the App:**
  - Incorporate advanced noise reduction techniques to improve performance in challenging audio conditions.
  - Optimize local models, particularly Whisper, for deployment on resource-constrained devices.
  - Implement real-time transcription for live captioning and other real-time applications.
  - Integrate speaker diarization to identify and label different speakers.
  - Develop robust methods for transcribing and representing mathematical equations and specialized notations.
- **Further Research:**
  - Explore new and advanced deep learning architectures for STT, such as Conformer models.
  - Integrate emotion recognition to provide a richer understanding of the spoken content.

The insights gained from this project provide a valuable roadmap for future research and development efforts in the rapidly evolving field of speech technology.

References

[1] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL]. URL: https://arxiv.org/abs/2006.11477.

[2] Nguyen Vu Le Binh. *nguyenvulebinh/wav2vec2-bartpho*. 2024. URL: https://huggingface.co/nguyenvulebinh/wav2vec2-bartpho.

[3] Thai-Binh Nguyen and Alexander Waibel. "Synthetic Conversations Improve Multi-Talker ASR". In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024, pp. 10461–10465. DOI: 10.1109/ICASSP48485.2024.10446589.

[4] Alec Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. 2022. arXiv: 2212.04356 [eess.AS]. URL: https://arxiv.org/abs/2212.04356.