

Graduation Thesis Final Report

*ENGLISH TO VIETNAMESE
SUBTITLE GENERATION SYSTEM*

Le Hoang Phuc

Ngo Anh Kiet

Kieu Minh Duy

**A thesis submitted in partial fulfillment of the degree of BSc. (Hons.) in Artificial
Intelligence with the supervision of Assoc. Prof. Phan Duy Hung**




**Bachelor of Computer Science
Hoa Lac Campus - FPT University
6th May 2023**

ACKNOWLEDGEMENT

This thesis is dedicated to our hard-working Assoc. Prof. Hung's much-needed assistance, continuous support, and funding made this work reach its fullest potential today. We would love to thank FPT University for always supporting us with precious courses to access resources to help us complete this thesis. Besides, we would love to thank the rest of our teammates for their effort and great contribution. We also would love to thank family and friends with encouragement. Without any of the support, this thesis would not have been completed.

DECLARATION

This thesis is the result of our work and includes nothing which is the outcome of work done in collaboration except where expressly indicated in the text. It has yet to be submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signed:    _____
Date: 26/04/2023 _____

Le Hoang Phuc, Ngo Anh Kiet, Kieu Minh Duy, and full qualifications
FPT University

ABSTRACT

Recently, the applications of artificial intelligence (AI) in many life domains are becoming increasingly popular and diverse. Besides that, AI appears in personal healthcare and education, automation in logistics and production, etc. Artificial intelligence applications in education have proven their role and capabilities, especially with access to up-to-date knowledge from texts, manuscripts, textbooks, lectures, and videos in foreign languages. Until now, most of the applications and models have worked well in English, Spanish, and other popular European languages. We have questioned whether we can create an application to help Vietnamese people expand their knowledge or simply understand foreigners' content. In this thesis, we create an application for generating Vietnamese subtitles from audio, video, and Youtube URL, and some sideline-related functions using well-perform models during the process. The application is divided into three main tasks: enhancement, recognition, and translation. After passing in, all background noise of input video or audio will be minimized, and human speech will be improved in the enhancement module before it can be processed in recognition. With recognition, our application uses an automatic speech recognition model to transcribe the spoken English language into text format. Then, the transcript text is passed into the translation module to be processed using a neural network and machine translation algorithm to create the Vietnamese subtitles. This thesis provides an effective solution for English-Vietnamese subtitle generation, and its outcomes demonstrate high accuracy and quality. In addition, a dataset is collected in order to evaluate models as well as services. Our validation dataset includes 40 hours of audio along with subtitles in Vietnamese and English. Among the chosen models tested on our dataset, Whisper-medium has shown a significant WER score of 1.0065. For the translation part, we have come up with a preprocessing method for the input text. Thanks to combining that method with the EnviT5 model, the BLEU score is slightly improved by minimum of 0.5% compared to the original and particularly shows its advantage over others such as Google Translate and Amazon Web Services.

Keywords: Natural Language Processing, Subtitle Generation, MTet, Artificial Intelligence.

CONTENTS

ACKNOWLEDGEMENT.....	1
DECLARATION.....	2
ABSTRACT.....	3
CONTENTS.....	4
LIST OF TABLES.....	5
LIST OF FIGURES.....	6
LIST OF ABBREVIATIONS AND ACRONYMS.....	7
1. INTRODUCTION.....	8
1.1. Motivation.....	8
1.2. Related works.....	9
1.2.1. Speech enhancement.....	9
1.2.2. Speech recognition.....	10
1.2.3. Machine translation.....	10
1.2.4. Final solution.....	11
1.3. Objectives & Contributions.....	12
2. DATA PREPARATION.....	13
3. METHODOLOGY.....	13
3.1. Application Overview.....	13
3.1.1. Recognition feature.....	13
3.1.2. Translation feature.....	14
3.1.3. Tubescribe feature.....	14
3.2. Model’s architecture.....	15
3.2.1. Overview.....	15
3.2.2. Enhancement.....	15
3.2.3. Recognition.....	17
3.2.4. Translation.....	18
4. EXPERIMENT AND EVALUATION.....	20
4.1. Experiment settings.....	20
4.2. Evaluation metrics.....	20
4.2.1. Word Error Rate Score (WER score).....	20
4.2.2. Bilingual Evaluation Understudy Score (BLEU score).....	21
4.3. Experimental comparison.....	22
4.3.1. Speech enhancement models comparison.....	22
4.3.2. Speech recognition models comparison.....	23
4.3.3. Machine Translation models comparison.....	23
4.4. Cost computation performance:.....	24
5. CONCLUSIONS AND FUTURE WORKS.....	24
APPENDIX A: USER INTERFACE.....	26
REFERENCES.....	29

LIST OF TABLES

Table 1: Architecture details of the Whisper model family.....	16
Table 2: WER score of speech recognition models (ascending order) evaluated on our dataset.....	22
Table 3: BLeU score of machine translation models (ascending order) evaluated on our dataset.....	22
Table 4: Average time execution for audio and video file of 60 seconds.....	23

LIST OF FIGURES

Figure 1: SpeechText.AI provides a comparison chart of their accuracy and other services. [1].....	8
Figure 2: Pipeline of Automatic Subtitle Generation Application.....	12
Figure 3: Recognition feature flow.....	13
Figure 4: Translation feature flow.....	14
Figure 5: Tubescribe feature flow.....	14
Figure 6: Causal Demucs with noisy input speech (bottom) and clean speech (top). Arrows represent U-net skip connections. [14].....	15
Figure 7: Encoder block’s structure of Demucs. [14].....	16
Figure 8: Decoder block’s structure of Demucs. [14].....	16
Figure 9: Overview of Whisper architecture. [17].....	18
Figure 10: Architecture of Transformers. [30] EnviT5 uses the same architecture with N=12.....	19
Figure 11: Google Colab free-tier’s GPU information.....	22
Figure 12: Interface of translation home page.....	25
Figure 13: Interface of recognition home page.....	26
Figure 14: Interface of Tubescribe homepage.....	26

LIST OF ABBREVIATIONS AND ACRONYMS

Abbreviations	Meaning
AI	Artificial Intelligence
API	Application Program Interface
ASR	Automatic Speech Recognition
AWS	Amazon Web Service
BLEU	Bilingual Evaluation Understudy
DBMS	Database Manager System
DEMUCS	Deep Extractor for Music Sources
GELU	Gaussian Error Linear Unit
GLU	Gated Linear Unit
LSTM	Long short-term memory
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
SNR	Signal-to-Noise Ratio
SRT	SubRip Subtitle File
TXT	Text File
URL	Uniform Resource Locator
WER	Word Error Rate
XML	Extensible Markup Language File
PESQ	Perceptual Evaluation of Speech Quality

1. INTRODUCTION

1.1. Motivation

Recently, generating subtitles has been a prevalent and extensive field related to natural language processing (NLP). Their applicability often appears in video processing, audio processing with background noise filtering, or real-time noise filtering, like generating subtitles for live stream platforms. The idea of this article originates from the increasing number of videos on social media serving many different purposes, from imparting knowledge, culture, or content in many areas like finance, health, and education. Human psychology is easily attracted by image or sound signals because the information transmitted through them makes academic ideas easier to understand and reach their imagination faster than the ideas in text form.

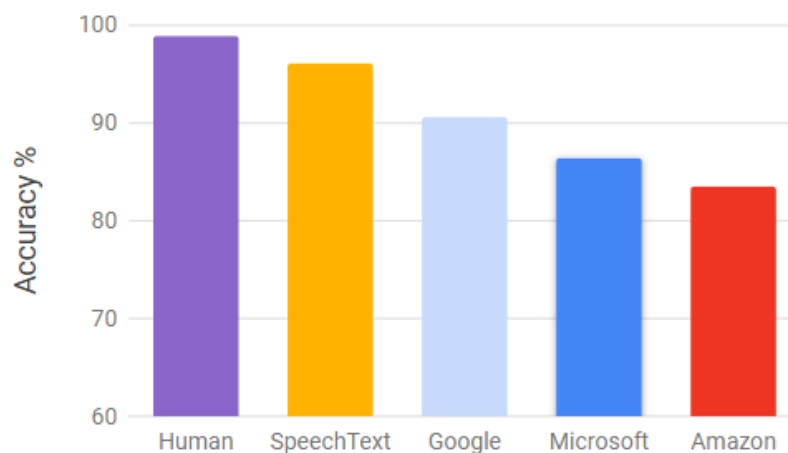


Figure 1: SpeechText.AI provides a comparison chart of their accuracy and other services. [1]

Not only mentioned in the world but even in Vietnam, the demand for creating subtitles is enormous. Almost official sources of information, the latest updates from science and technology to finance and banking, political situations, etc., come from English websites. Therefore, users will have some significant troubles. First, searching and researching requires a significant amount of English knowledge. At the same time, foreign language skills are a set of techniques that users must take a long time to cultivate regularly, which is not easy for them to master in a short period. Second, third-party applications or services providing solutions for subtitles are too hard to access. If any, they will be user-contributed subtitles, the automatic Vietnamese subtitles translated incorrectly, or services that must be paid a fee. Not all Vietnamese people can meet the requirements to access services like that.

To tackle these problems, there are many solutions proposed, and one of them is an automatic subtitle generation system. Almost services providing speech transcription, like SpeechText.AI [2], Google Cloud Speech [3], Scale [4], Rev.ai [5], or Amazon Transcribe [6], have the same speech-to-text capability (with different accuracy). Besides, some services provide extra functions (text summary, a time-stamp for each word, and a unique

script for each speaker). Although SpeechText.AI [2] shows higher accuracy than other services (Figure 1), they offer high pricing (the free tier is limited compared to other services like AWS). In addition, for now, we have found an online video editing website named Veed.io [7] that provides auto-transcribe and auto-translate for videos uploaded by users. However, there is an additional fee for using their features and it is quite expensive for ordinary user. Until now, there are no extensions, applications, or services that support generating transcription in Vietnamese directly with a Youtube URL. About Youtube, they allow users to use some built-in features for automatic subtitle generation and automatic translation. Although there are no publications about how it works and the usage model, they have recommended that these features' quality may vary and users should review and edit the subtitle file themselves [8]. In addition, there are a few articles noting that these built-in features do have poor accuracy [9].

Many services and techniques are applied to generate transcription, or translate from one language to another and vice versa. Depending on aspects like the ability to pay for using services and the trade-off between the hardware available and the state-of-the-art model, this thesis has focused on comparing the latest optimized algorithms, models, and services, and finally combining them to create a robust application for generating Vietnamese subtitles from a random video, especially easily obtained by a part of Vietnamese users.

1.2. Related works

1.2.1. Speech enhancement

According to J. Benesty, S. Makino, and J. Chen, speech enhancement means “improving the intelligibility and quality of a degraded speech signal” [10] captured under noisy or degraded conditions. Over time, many different methods have been researched and proposed, and at each period, they all showed their advantages. In 2014, Kaladharan N. stated that “Speech signal is always accompanied by some background noises” [11]. He then proposed a speech enhancement method based on the spectral subtraction algorithm and evaluated it by the signal-to-noise ratio (SNR). The result is quite positive but he mentioned that this method is not well-performed in blare situations. Another method is proposed by Chaogang et. al. based on a combination of Karman filtering, spectral subtraction, and Liljencrants–Fant (LF) excitation [12]. This combination gave higher SRN and PESQ results than either Karman filtering or spectral subtraction individually. However, the essence of this method is still to apply the pure signal processing method, so it must be based on a set of parameters to be able to apply to the formula to give a proper result. So maybe this parameter set may be good for one environment but not for another, hence its generality is not guaranteed. The same problem occurred in the magnitude and phase spectrum compensation method proposed by Zhen L. et. al. [13] when they experimented that $\lambda=3.74$ maximized both PESQ and SNR scores in their evaluation.

In recent years, the trend of using deep learning has become more and more popular. New models were born one after another, and the later models showed

superior results compared to the previous ones. Gradually, deep learning models have shown their generalization and high applicability. Their approach varies from raw waveform [14] to Mel spectrogram [15].

After reviewing, we have divided speech enhancement techniques into two categories: traditional and deep learning. The most common traditional approaches are spectral subtraction, Kalman filtering, magnitude, and phase spectrum compensation as we mentioned above. Although traditional methods have the advantages of processing time, simple implementation, and hardware savings, they can not be used in various environments or perform poorly (such as in non-stationary noise conditions in the real world). However, with the original purpose of finding a mapping function between noisy and clean speech, deep learning has shown its generalization capability and ability to work properly in almost any condition (depending on the training dataset) since the first time it was proposed in 2015 [16] until now. Thus, following the track of deep learning is promising.

1.2.2. Speech recognition

Along with the innovations of hardware, speech recognition has several waves of major innovations, the most recent, this field has benefited from advances in deep learning and big data. From the first-ever application of the Audrey system that can only recognize digits to IBM's Shobox, which understood and responded to 16 words in English, now it can do a lot of complex tasks (such as in-car systems, military, healthcare, etc.).

For our application, speech recognition plays an important role. It affects the result of the next step, i.e., translation, which is the final goal. In recent years, large technology corporations have published models with surprising performance, such as OpenAI Whisper [17], Facebook's wav2vec [18], and wav2vec 2.0 [19]. With both two Facebook models, all recognize speech in upper case and punctuations are removed. In other words, the output of wav2vec2s only contains upper texts only, in which we can not distinguish proper nouns and normal words. The listed models provide a significantly low WER score result on the LibriSpeech dataset [20]. Especially, Whisper also automatically generates sentence punctuations for output text. However, the long overlapping conversations are not supported by these models.

1.2.3. Machine translation

In translation, or machine translation in particular, context plays an essential role in transitioning from one language to another. One word in this language can have multiple meanings in another language(s) based on the context of a paragraph or a complete conversation. The lack of context will cause the accuracy of the entire translation application to deteriorate drastically. Also due to the poverty of information, most of the translated captions have to be made by humans and clearly human involvement makes these processes ineffective for both the content creators and the users. Amazon's service [21] translates the entire conversation using speech-to-text models that support sentence separation. Including dialogue in

sentences rather than words, helps improve contextualization in machine translation. Nevertheless, the problem with Amazon Translate Service is that their model is a closed-source model, meaning that users can only use it through the API. The use of the API has an inherent disadvantage in that users can not control or improve model quality, and the price of the AWS is not suitable for many common users. Most importantly, neither Amazon's nor Google's machine translation model has high accuracy, and this will reduce the reliability of the entire application. Ultimately Meta's subtitle translation model [22] seems to be the best of the approaches above when Meta uses a full sentence translation method by incorporating a model of speech recognition. But they do not have tools such as APIs, open-source code, or documentation describing specifically the algorithm behind their automatically translated caption generator to help developers apply the process to other problems. Recently, along with the increasing perfection of the Nature Language Processing model, VietAI has introduced a state-of-the-art model EnViT5 [23] for the problem of machine translation between Vietnamese and English. The EnViT5 model is based on Transformer Architecture along with using the method proposed by T5 [24]. VietAI is also developing their own trainable separate Vietnamese-English dataset for the fine-tuning stage. Combining all the processes led their model to achieve the highest results in the English-to-Vietnamese machine translation task. Also because this is an open-source model, it means more straightforward in the testing phase and adjusting parameters, which in turn speeds up the process of applying the model to other machine translation tasks including automatically translated caption generator.

1.2.4. Final solution

After consulting the system's design of some services from AWS, Youtube, and Facebook's automatic captioning, etc., we decided to divide the automatic subtitle generation application into two main sub-tasks, as shown in Figure 2 below. The input can be audio, video, or a Youtube URL depending on each function. Based on the input type, the application has separate processing steps for them. After the audio is loaded, it will go through a recognition module, which will be enhanced and recognized. In the next step, the output of the recognition module will be translated. Finally, our application does the final process steps and exports output files to the user.

About the backbone, we use a deep learning approach in the entire pipeline. Specifically, when considering some aspects like accuracy, and capability of deploying, a model named Demucs is chosen to improve the quality of speech input in the enhancement phase in the recognition module. Another phase in the recognition module, speech recognition uses the Whisper model of OpenAI, and the Translation module leverages a model called EnViT5 built by VietAI, a Vietnam academic research organization. In addition, we also dig deep into those models to find out the way to minimize their limitations and generate the most suitable outcomes, more details about our processing method will be discussed in Section 3.2.4. To sum up, our application proposes the data processing methods while utilizing the best performance

models to achieve the desired outcomes. The comparison between models and their experimental results is shown in Section 4.

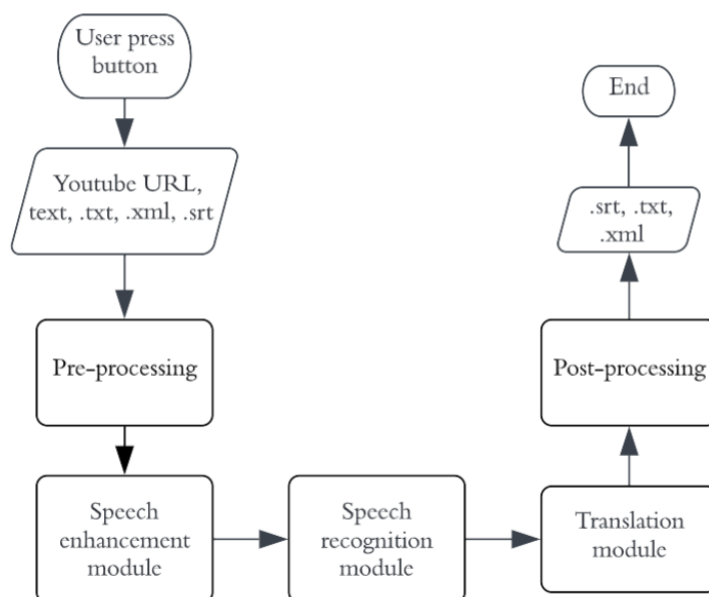


Figure 2: Pipeline of Automatic Subtitle Generation Application.

1.3. Objectives & Contributions

The main goal of this thesis is to create the N2Vi - English to Vietnamese subtitle generation application that provides ease of access for Vietnamese users. This thesis results in an effective pipeline for creating Vietnamese subtitles based on speech processing, and the latest models to achieve this goal. Our approach builds upon previous developments and leverages deep neural networks with transfer learning and attention mechanisms to improve the quality of speech signals. Besides, we also collected a dataset and used it to evaluate some recent state-of-the-art models and then come up with the most suitable ones for our application. Last but not least, we also dig down into those models to carry out processing methods that can improve the final outputs.

2. DATA PREPARATION

About data collection, data is crawled from Youtube platforms to evaluate the effectiveness of chosen models. The dataset consists of more than 40-hour video content with both English and Vietnamese transcriptions. We employed a web crawling technique supported by Youtube API [25]. to extract and select the desired video data from Youtube to gather this dataset. Youtube has many videos that allow us to collect a large and diverse set of videos with various speakers, accents, background environments, and topics, enabling us to experiment with models' performance in generating high-quality subtitles. The collected data would also help us to test the robustness and accuracy of our English to Vietnamese subtitle generation application.

We filter videos based on whether that video supports song language subtitles. If the video contains Vietnamese-English subtitles, then it will be selected to download to our database. The data to be downloaded will include one audio file (MP4 format) and two subtitles files in English and Vietnamese (XML format). Those youtube channels with Vietnamese-English bilingual subtitles are infinitely few, so the data of our group is taken mainly on the Ted-ed channel.

3. METHODOLOGY

3.1. Application Overview

Besides our main fully functional application, we also implement sideline-related functions: recognition and translation. We have designed their flow, and they will be elucidated below.

3.1.1. Recognition feature

Figure 3 shows the flow of our stand-alone recognition feature. At first, our webpage takes the input file from users, which is formatted in “.mp3”, “.wav”, “.mp4”, and “.m4a”. If the input is a video file, audio will be extracted in the pre-processing step. After that, 16kHz loaded audio will be enhanced and then passed into the recognition module. We give the user two options in this feature, i.e., recognizing the input file in English or Vietnamese. If the user wants to get Vietnamese content out of the input file, the translation module will get in charge. After post-processing steps and file exportation, the user will be able to preview the content of the input file and available “.txt”, “.str”, and “.xml” files to download. Based on the user's selection, these output files contain either English or Vietnamese content.

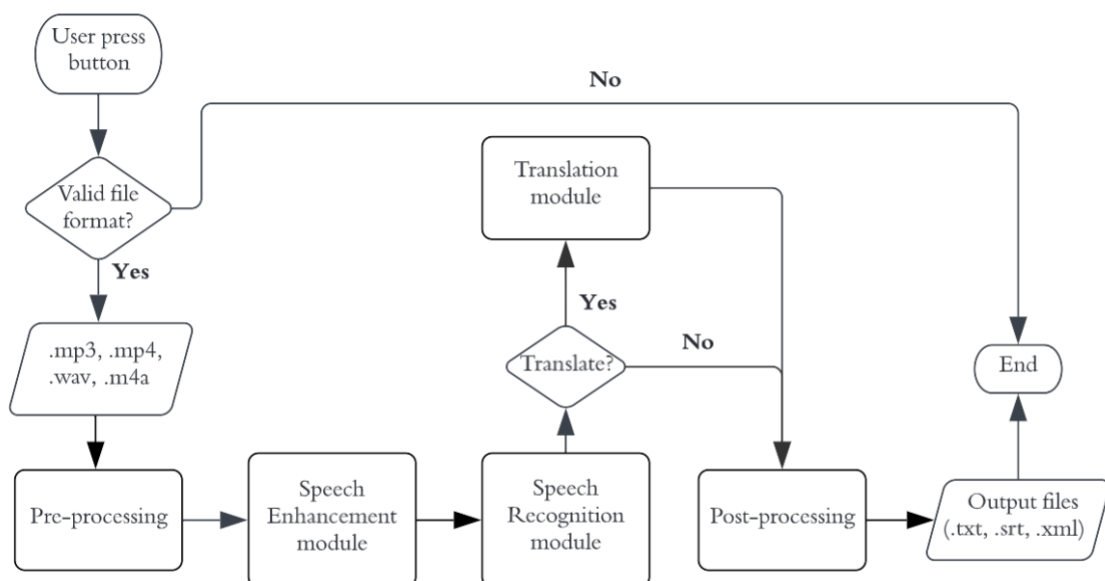


Figure 3: Recognition feature flow.

3.1.2. Translation feature

The translation feature, alone, is quite simple, as shown in Figure 4 above. The user can upload a “.txt”, “.str”, or “.xml” file to our webpage, then they must choose what language is used in this file. In the next step, our application will pre-process and put the extracted texts into the translation module to return the translated texts. These texts then will be processed in post-processing and exported. One notation is that when the user uploads the “.txt” file, the translation module will only return the “.txt” file. For other file formats, both “.txt”, “.str”, and “.xml” are available.

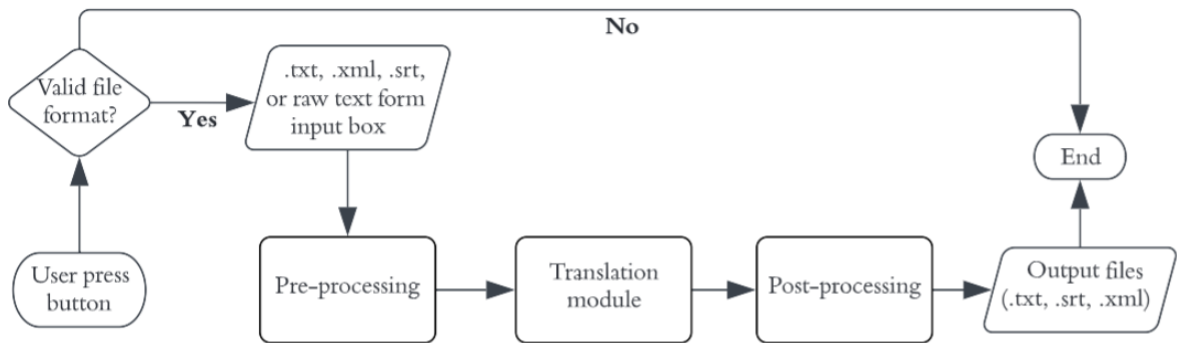


Figure 4: Translation feature flow.

3.1.3. Tubescribe feature

Last but not least, an application designed to help users in Vietnam access content from a seemingly endless source of videos from Youtube, Tubescribe (Figure 5). “Tubescribe” is a short-term Youtube scribe. End-user only needs to paste the Youtube video URL they want to transcribe to the input box. After that, the URL is checked for validation, then the video's audio will be downloaded by the pytube package [26] and loaded. After ensuring that the audio is loaded and pre-processed, N2Vi will transmit

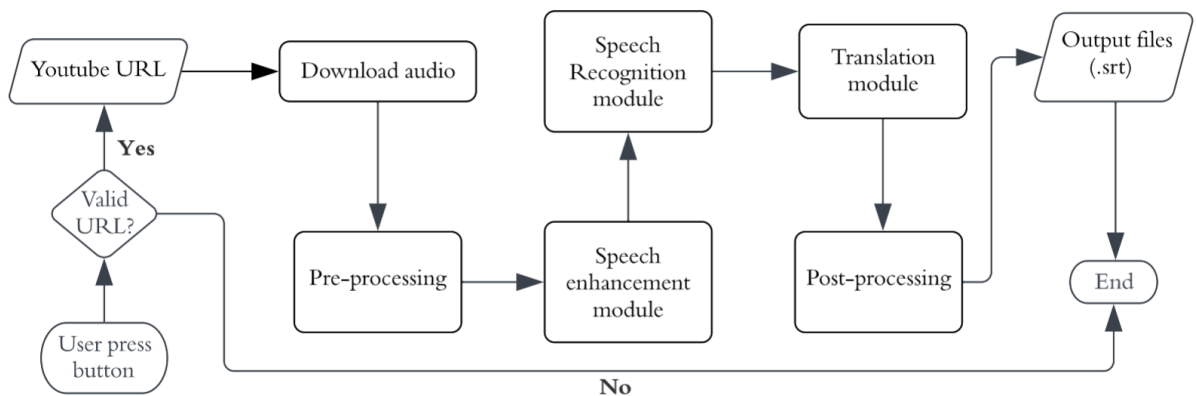


Figure 5: Tubescribe feature flow.

to the speech enhancement module. Once the enhancement is complete, its output will be recognized and then passed into the translation module. The “.srt” file can be downloaded when the process is done. If users then want to watch Youtube videos with Vietnamese subtitles, we recommend they install a third-party extension to their

browser named “Subtitles for Youtube” [27] and then upload the application's “.srt” output file.

3.2. Model’s architecture

3.2.1. Overview

Based on the original purpose of our application, we have experimented with several models for each relevant task and end-to-end model. After analysis and evaluation, we applied three of the most suitable transformer models for our NLP-related application. The remaining sections of 3.2 explain the architecture of the most suitable models we use in our application, and we have experimented on several models for each task. The comparison between those is discussed in section 4.2 to inform why we use the following models.

3.2.2. Enhancement

As we briefly mentioned, a pre-trained encoder-decoder architecture called Demucs will enhance human speech and minimize background noise. The overview of the architecture of Demucs [14] is shown in Figure 6.

One noteworthy improvement of Demucs is that it uses transposed convolutions rather than linear interpolation followed by a convolution with a stride of 1, as

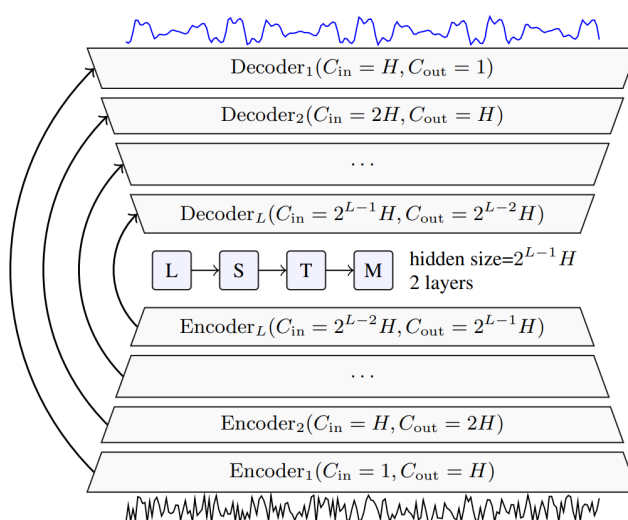


Figure 6: Causal Demucs with noisy input speech (bottom) and clean speech (top). Arrows represent U-net skip connections. [14]

implemented in Wave-U-Net [28]. These transposed convolutions require four times fewer operations and memory, which means more channels will be used, and the model results in better outcomes. To explain this architecture further, we add Figure 7 and Figure 8 to describe encoders and decoders visually. The first encoder layer receives raw waveform as input and outputs a latent representation.

The output will be fed to the next encoder block or a sequence model (LSTM). Each encoder consists of a convolution layer with a kernel size of K and stride of S

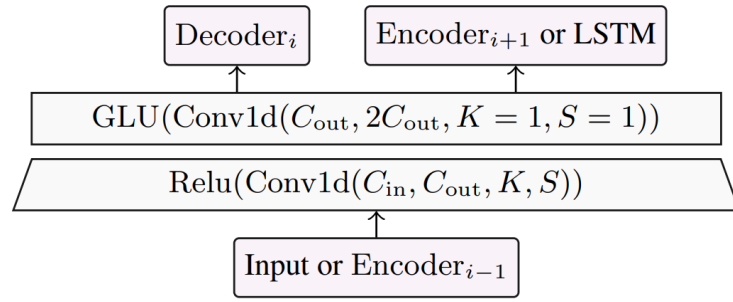


Figure 7: Encoder block's structure of Demucs. [14]

with $2^{i-1}H$ output channels, followed by a ReLU activation, a “1x1” convolution with 2^iH output channels, and finally, a GLU activation that converts back the number of channels to $2^{i-1}H$ (Figure 7). Next, LSTM takes the latent representation from the last encoder as an input, and outputs a non-linear transformation of the exact size. This layer comprises two layers and $2^{L-1}H$ hidden unit(s). After that, the output of LSTM is passed to the decoder block, which returns an estimation of clean speech. The encoder layer i^{th} of the decoder takes as input $2^{i-1}H$ channels and applies a 1x1 convolution with 2^iH channels, followed by a GLU activation function that outputs $2^{i-1}H$. Finally, a transposed convolution with a kernel size of 8, a stride of 4, and $2^{i-2}H$ output channels accompanied by a ReLU function. The output for the last layer is a single channel with no ReLU (Figure 8).

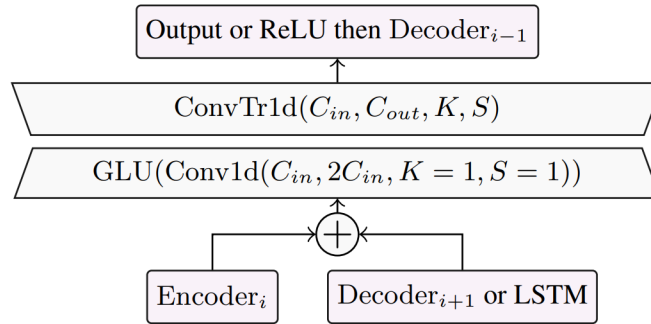


Figure 8: Decoder block's structure of Demucs. [14]

3.2.3. Recognition

For speech recognition, we used a pre-trained transformer model called the Whisper model [17] proposed by OpenAI. Whisper is an automatic speech recognition (ASR) built on an encoder-decoder transformer architecture. It is trained on 680,000 hours of multilingual and multitasking supervised data collected from the web. In the beginning, the input audio is split into 30-second chunks and converted into a log-Mel spectrogram [15]. After that, the spectrogram is passed into an encoder. The decoder has the ability to predict the corresponding text caption based on input audio, intermixed with special tokens that direct the single model to perform multiple tasks

such as language identification, phrase-level timestamps, and multilingual speech transcription.

According to OpenAI’s experimental result publication, they have trained five Whisper models with a different number of layers, heads, widths, and parameters from “tiny” to “large”. Our application inherits Whisper-medium due to its good

Table 1: Architecture details of the Whisper model family.

Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

performance and acceptable hardware requirements. This version consists of 24 layers, a width of 1024, 16 heads, and 769 million parameters (Table 1).

Deeper dive into Whisper, the complete architecture is described in Figure 9. In the beginning, the raw audio file will be loaded and transformed into an 80-channel log-magnitude Mel spectrogram representation. After converting, the encoder handles this Mel with two convolution layers and a GELU activation function. In the next phase, the encoder Transformer blocks are applied after Sinusoidal position embeddings are added to the GELU layer’s output. The transformer uses pre-activation residual blocks, and a final layer normalization is applied to the encoder output. The decoder then uses learned position embeddings and tied input-output token representations. One notation is that the encoder and decoder have the same number of transformer blocks [29].

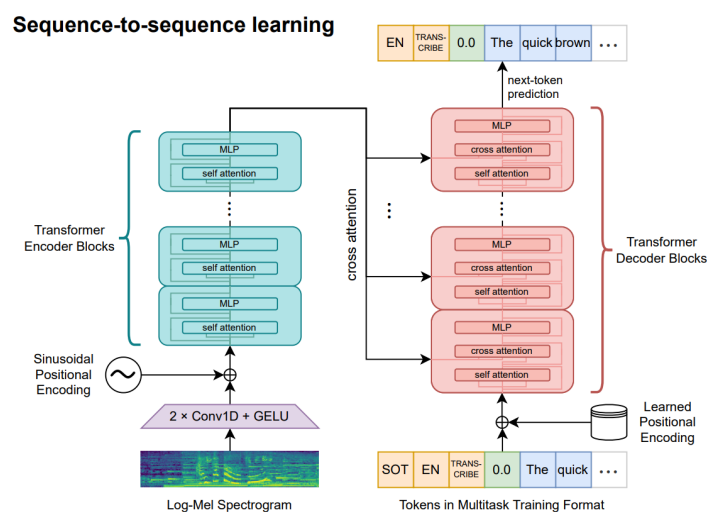


Figure 9: Overview of Whisper architecture. [17]

3.2.4. Translation

In the translation module, we use a pre-trained T5 (base) trained on the MTet dataset proposed by VietAI called EnViT5 [23] as a model backbone. EnViT5 is a Text-to-Text Transfer Transformer model that inherits and combines the proposed encoder-decoder architecture by Vaswani et al. [30] and the T5 framework of Raffel et al. [24] EnViT5 uses the base architecture config of the T5 practical purpose of their study, which consists of 12 encoder blocks and 12 decoder blocks. T5-Base is a variant of the T5 architecture which is built from 220 million parameters for transfer learning on a wide range of NLP tasks.

As the original transformer, each encoder block consists of multi-head attention, normalization, and a feed-forward layer with residual connection (Figure 10 - left side). After encoding, the output will be fed into decoder blocks. Again, two multi-head attention layers (masked attention and normal attention), normalization, and feed-forward layers take part in and handle the encoder's outputs. The output of the last decoder block is then passed to a linear layer; after that, a softmax is obtained to get the final output's probabilities (Figure 10 - right side). The concrete model of transformers is visually described in Figure 10, the EnviT5 uses the same architecture with $N=12$.

In the course of testing, we found that the EnViT5 model is susceptible to output errors when it produces the iterative output, this problem can occur in two cases: when the model has to infer text data that is too long (more than 512 characters) or when the required maximum output length is too large. This makes the accuracy of the model deficient. As we learned further, we found that the model does not support dividing long sentences for inference, so we devised a solution for processing input texts. We divide sentences that are too long into chunks so that the length of each chunk does not exceed the maximum generating output of the model, the default threshold is 256. This method will prevent long sentences from producing iterative outputs while retaining the context of sentences. Another problem is that when tokenizing special characters such as punctuations, the model does not retain the newline character (“\n”), and, more surprisingly, the Transformers framework does not support adding special characters to the tokenize model, which led our team to devise another process text method which, along with dividing the text into chunks, combines newline-based text division and joins the entire chunks and lines to produce the final result. By using only these two pre-processing methods, we were able to increase the Bilingual Evaluation Understudy (BLEU) score of the model to at least 0.5% and lead to the improvement of the entire application.

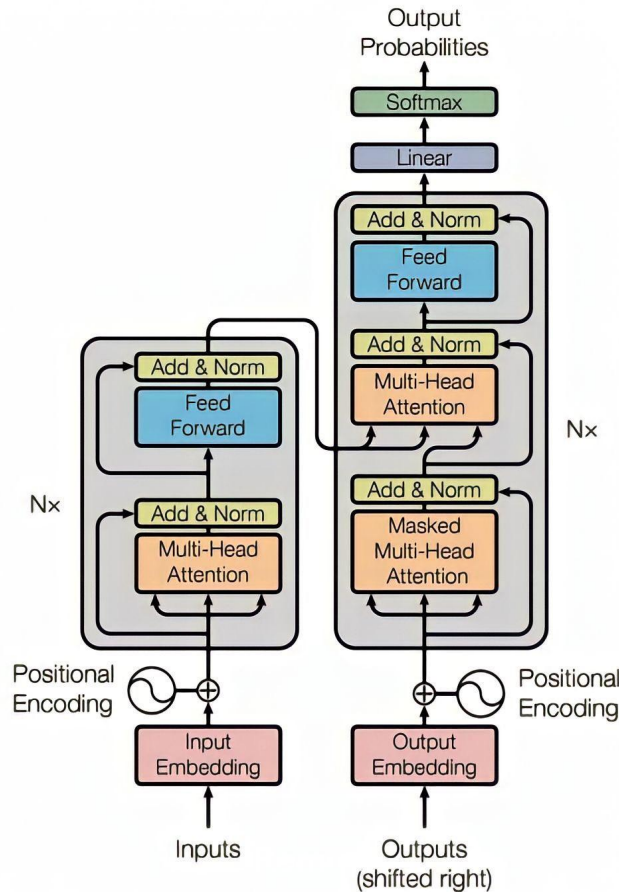


Figure 10: Architecture of Transformers. [30] EnviT5 uses the same architecture with $N=12$.

4. EXPERIMENT AND EVALUATION

4.1. Experiment settings

For our application's complete functionality, we have collected and created a dataset of 453 videos. All of these videos are crawled from Youtube, their length varies from 4 to 7 minutes, and contain conversations, reports, lectures, etc., from diverse fields, accents, genders, and ages. Also, we have 453 bilingual subtitle files in XML format, which will be used for evaluating machine translation tasks.

Audio is loaded as 16kHz mono audio. We randomly add White Gaussian noise to the input audio to ensure the denoise module quality. Since the result of the speech enhancement module can not evaluate directly, outputs are then processed by a speech recognition model to generate their transcriptions. After that, we assess the transcriptions with evaluation metrics. In finding a suitable ASR model, we experiment with and evaluate five models. After loaded, the audio is split into multiple chunks of 30 seconds and then processed by the ASR model. In the last module, translation, we compare two state-of-the-art Vietnamese models with the original subtitles crawled from Youtube. Each subtitle crawled is in the form of XML; we then transform it to JSON for convenience and accessibility. Due to the limitation of the translation model when inferring long sentences,

we created a function to split a single sentence into chunks with a threshold of 256 maximum characters per chunk.

4.2. Evaluation metrics

To evaluate our application's functionality, we use the two most common metrics in the field of NLP, i.e., WER score and BLEU score. A further explanation is described in detail in the following sections.

4.2.1. Word Error Rate Score (WER score)

WER is an important, common metric used to measure the performance of the ASR models/systems. WER is based on the "Levenshtein distance", which measures the differences between two strings. Scientists, developers, and others who use ASR technology may consider WER when choosing a service or a model for a specific purpose. ASR developers may also calculate and track WER over time to measure how their ASR model/system has improved. In simple terms, WER is used when deciding whether an ASR model is good enough.

Specifically, WER is the ratio of errors in a transcript to the total words spoken. The WER score equals the total of insertions (I), deletions (D), and substitutions (S) over the number of words in the reference/actual script (N) as written in (1). The number of words in the reference is the sum of the number of substitutions, deletions, and corrections. Each factor is defined:

- A substitution (S) occurs when a word gets replaced.
- An insertion (I) occurs when a word is added that is not in the ground truth
- A deletion (D) happens when a word is left out of the transcript completely

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

The lower the WER in the speech-to-text system, the better the accuracy in recognizing speech. Therefore word accuracy can be calculated by the formula in (2).

$$\text{WAcc} = 1 - \text{WER} \quad (2)$$

Even when WER is commonly used, it may not be appropriate for all ASR system/model evaluations due to recording hardware and environmental quality (low-quality microphone, noisy background, etc.) and some unusual proper nouns. However, with N2Vi, the enhancement collaborated with ASR to return textual transcripts. Thus, this metric is good to use.

4.2.2. Bilingual Evaluation Understudy Score (BLEU score)

The BLEU score is a metric used to evaluate the quality of machine translation output by comparing it to one or more human translations. It was introduced by Papineni et al. [31]. The BLEU score ranges from 0 to 1, where 1 indicates a perfect match between machine and human reference translations. It is calculated based on the n-gram overlap between the machine and reference translations, where n can be any

positive integer. The BLEU score is computed by taking the geometric mean of the n-gram precisions. The precision of each n-gram size is calculated as the ratio of the number of matching n-grams in the machine translation to the total number of n-grams.

The BLEU score formula calculates the geometric mean of the n-gram precision scores, where n ranges from 1 to a maximum value, typically 4. The formula is as follows:

$$\text{BLEU} = BP * \exp\left(\sum_{n=1}^N w_n * \log(p_n)\right) \quad (3)$$

where:

- *BP* (brevity penalty) is a factor used to penalize translations that are shorter than the reference translations. It is defined as follows:
 - If *MT* (machine translation) length == *Ref* (reference translation) length, *BP* = 1
 - If *Me precision sT* length > *Ref* length, $BP = \exp\left(1 - \frac{Ref}{MT}\right)$
- *N* is the maximum n-gram order to consider.
- *w_n* is a weight assigned to the n-gram order (usually set to $\frac{1}{N}$)
- *p_n* is the score for n-gram order n, defined as

$$p_n = \frac{\text{count_ngram_MT}}{\text{count_ngram_clip}} \quad (4)$$

where:

- *count_ngram_MT* is the number of n-grams in the MT that appear in the reference translations.
- *count_ngram_clip* is the maximum number of times each n-gram appears in any reference translation.

Note that the n-gram precision score is clipped to a maximum value of 1 to avoid penalizing translations with more matches than the maximum number of occurrences in the reference translations.

4.3. Experimental comparison

4.3.1. Speech enhancement models comparison

We have found some models from SpeechBrain and Facebook AI research. Due to the limitation of hardware conditions from Google Colab (we used this platform for running and testing models) with 12.7 GB RAM and approximately 15GB GPU RAM (Figure 11), when running 2 model versions from SpeechBrain, our session crashed because CUDA was out of memory. Therefore, in our application's first-ever version, we must use Facebook's Denoiser (Demucs) in the enhancement module and then

improve and optimize it in later versions. However, we have calculated the WER score of Denoiser by taking its output and feed to the recognition model (Whisper-medium). The result is quite good when the WER score is 0.1922 on our dataset.

```

1 !nvidia-smi

Thu Apr 13 09:55:01 2023

+-----+
| NVIDIA-SMI 525.85.12      Driver Version: 525.85.12   CUDA Version: 12.0     |
+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+
| 0   Tesla T4      Off          | 00000000:00:04:0 Off |    0         0         0 |
| N/A   60C    P8     10W /  70W   |  0MiB / 15360MiB |      0%      Default  |
+-----+-----+-----+

```

Figure 11: Google Colab free-tier's GPU information.

4.3.2. Speech recognition models comparison

Before choosing which model we can implement with the speech recognition module, we have evaluated five models with our prepared dataset, as we mentioned. We chose the following models because they are either the most downloaded or state-of-the-art in the NLP domain. Some of them satisfy the punctuation and proper name requirements. On average, each of them cost 6 to 8 hours of running.

With two versions of Facebook, although the size of wav2vec2-large is almost 3.5 times larger than wav2vec2-base, the result is just slightly different (0.0022). Jonasgrosman successfully fine-tuned his/her model from wav2vec2-large when its WER score was ~ 0.03 points better. The result outperforms the rest in the top 2 Whisper models, especially the Whisper-medium. The Whisper-tiny is built up with only half the number of parameters of Facebook's wav2vec2, its result is even better.

Whisper also generates punctuations and auto-capitalizes proper nouns for its output besides a good WER, as shown in Table 2. Thus we concluded that the Whisper medium is appropriate for our N2Vi expectation result.

Table 2: WER score of speech recognition models (ascending order) evaluated on our dataset.

<i>Model name</i>	<i>Params</i>	<i>WER</i>
openai/whisper-medium	769M	1.0065
openai/whisper-tiny.en	39M	1.0312
jonatagrosman/wav2vec2-large-xlsr-53-english	315M	1.0327
facebook/wav2vec2-base-960h	94M	1.0766
facebook/wav2vec2-large-960h-lv60-self	315M	1.0788

4.3.3. Machine Translation models comparison

There are not many English-Vietnamese machine translation open-source models, the most popular way is using a third-party's machine translation model through API,

Table 3: BLeU score of machine translation models (ascending order) evaluated on our dataset.

<i>Model name</i>	<i>BLeU</i>
Google translate	0.2453
Amazon translate	0.2969
EnViT5-base	0.3192
EnViT5-base + Our preprocessing method	0.3255

for instance, Google Cloud, AWS, etc. Therefore, we use the BLeU score to compare the EnViT5 model with our pre-processing method against the original model and these cloud engine machine translation models (Table 3). Surprisingly even when using a simple technique in the preprocessing stage, it can still increase the final BLEU score slightly to 0.3255.

4.4. Cost computation performance:

Our application will eventually be deployed as a web application. This web application allows users to easily interact with drag, drop the inputs, and download its output. On average, N2Vi takes about 41 seconds to start up on the first run (for the server side). After startup, users can access and use it directly through the URL. We tried to infer an audio file. The application needs 16 seconds to complete the recognition step. Then, if there is any request to translate the recognition's output, the application runs for another 3 seconds. For a video file, it took 57 seconds to recognize and 2 more seconds to translate (Table 4). The time for recognizing a video file is unusually long because it is partly affected by the network's transmission speed.

Table 4: Average time execution for audio and video files of 60 seconds.

<i>Module</i>	<i>Latency (s)</i>	
	<i>Audio file</i>	<i>Video file</i>
Translation	2	2
Recognition	16	57
Recognition in Vietnamese (Recognize + Translate)	18	58

For Tubedownload, because the input is an URL and we then need to download audio from the Youtube source, time execution for this function may vary according to the network's transmission speed. But overall, the time measured for recognition and translation approximates the one calculated for audio (tested on an URL of 1 minute Youtube video).

5. CONCLUSIONS AND FUTURE WORKS

In this study, we have learned about the relevant tasks of a Vietnamese subtitle generation application. We have tested, compared, and improved in inferring those pre-trained models. At the same time, an end-to-end “N2Vi” captioning application is designed, connected, and implemented. Up to the present, our application can mostly meet users' basic usage needs and the expectation for great improvements in the future.

N2Vi, yet, can only process English-language audio/video inputs with small overlapping speakers' speech. In future works, we would like to release a more robust application. Specifically, future versions will be able to operate faster, serve as an API and Chrome extension, and, most importantly, it is available to perform in various input languages in real-time so that people can use it directly in conversations, conferences, or lectures.

APPENDIX A: USER INTERFACE

1. Overview

For the user interface, we have customized icons, backgrounds, motions, etc., to be more user-friendly. The back-end was developed based on the Flask framework to connect modules, functions, and data processing to return the result back to the front-end.

In order to communicate between a front-end and a back-end, we use the Ajax method in JQuery, which allows the page not to have to reload when it receives a back-end processing data and speeds up the communication between two processes.

2. Translation

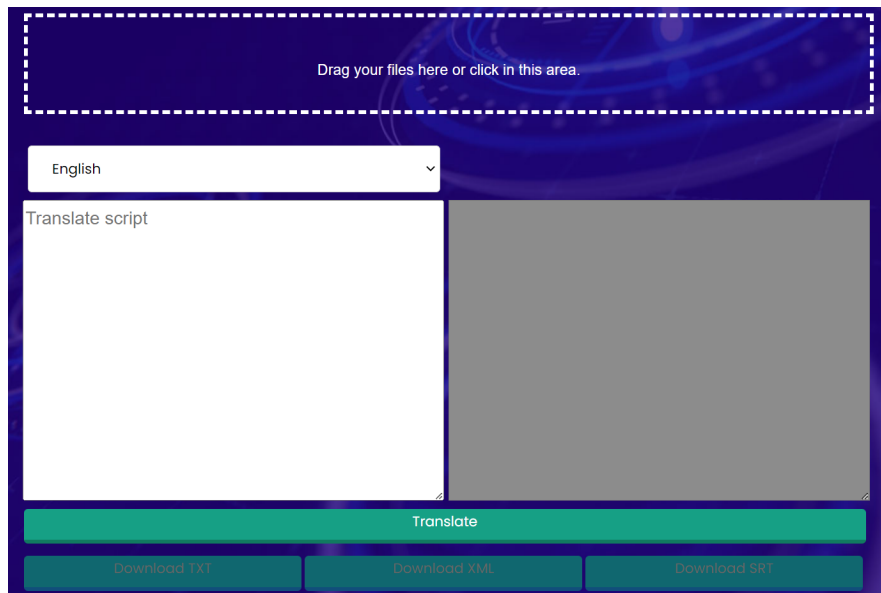


Figure 12: Interface of translation home page.

Since the primary goal of the translation module is not merely to translate word-by-word but also to translate video subtitle files like “.srt” or “.xml”, therefore our translation feature supports uploading “.txt”, “.xml”, and “.srt” files. Users are able to choose between Vietnamese or English as the source language, and users must choose so that the model can give more accurate results. When our server receives a file from the user, the “Translate script” box (the left white box under the language box) will display its content for users to check and edit the file before translation (Figure 12). Recently, most video editing applications now use “.srt” files to generate subtitles, so this feature will prioritize returning “.srt” files at the end of the process. The user can view the translated contents in the right darker box and select the file format to download at the bottom of the page.

3. Recognition

In the recognition feature (Figure 13), users can upload audio files in the following formats: “.mp3”, “.mp4”, “.m4a”, and “.wav”. Due to hardware limitations and system overload

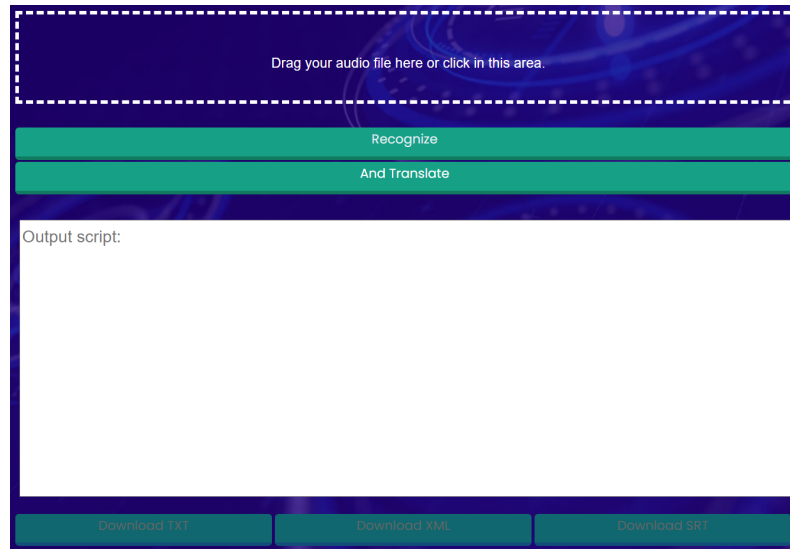


Figure 13: Interface of recognition home page.

potential, we limit the size of files that users can upload to less than 10 Mb. Current Vietnamese recognition models are not very good, they do not support sentence separation, and mostly they are word-by-word models. This means that if using these models, their results in subtitles form will be inaccurate translation lies to their lack of context. Most importantly, word-by-word subtitles will be more difficult to insert into videos because most subtitle applications currently only support sentence-by-line subtitles, not word subtitles. Therefore, currently, our website only supports recognizing English language inputs. Once the audio file is uploaded, users will have two options: recognize the audio file in English, or in Vietnamese as the final result. Then the output will be displayed in the box so that the user can preview the output before downloading. Similar to the translation process, users will also have three options for output file formats: “.txt”, “.xml”, and “.srt”.

4. Tubscribe

Tubscribe is a combination of recognition and translation, which aims to make it easier for users to access videos on YouTube without facing language boundaries. As described in Figure 14, Users only need to enter the URL of the desired YouTube video and then we use the YouTube API to check if the size of the video exceeds the processing limit, otherwise, the video will be saved to the database as an audio file and run through the following steps: denoise, recognize, translate into Vietnamese, and finally export to a “.srt” file. Users can download the “.srt” file and use some third-party software (like extensions) to view the subtitles on the desired Youtube video.

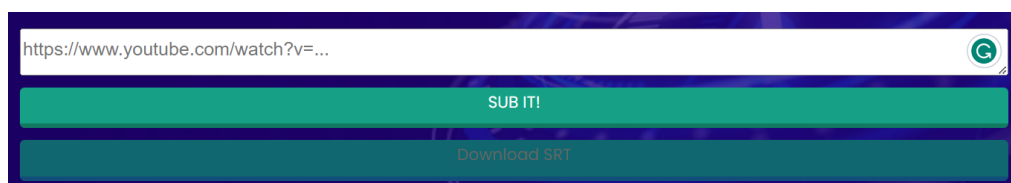


Figure 14: Interface of Tubscribe homepage.

REFERENCES

- [1] AI Transcription Service. (n.d.). SpeechText.AI. Retrieved April 17, 2023, from <https://speechtext.ai/>.
- [2] SpeechText.AI. (2020). Speech Recognition API Reference. Retrieved from <https://speechtext.ai/speech-api-docs>.
- [3] Google Cloud. (n.d.). Speech-to-Text documentation. Retrieved from <https://cloud.google.com/speech-to-text/docs/>.
- [4] Scale. (n.d.). Guides and quickstarts for integrating Scale products. Retrieved from <https://scale.com/docs>.
- [5] Rev. (n.d.). Documentation. Retrieved from <https://www.rev.com/api/docs>.
- [6] Amazon AWS. (n.d.). Amazon Transcribe API Reference. Retrieved from <https://docs.aws.amazon.com/transcribe/latest/APIReference/Welcome.html>.
- [7] VEED.IO. (n.d.). Pricing. Retrieved May 11, 2023, from <https://www.veed.io/pricing>.
- [8] YouTube. (2023). Use automatic captioning. In YouTube Help. Retrieved April 17, 2023, from <https://support.google.com/youtube/answer/6373554?hl=en->.
- [9] UMN Duluth. (n.d.). Correcting YouTube Auto-Captions. Retrieved May 11, 2023, from <https://itss.d.umn.edu/centers-locations/media-hub/media-accessibility-services/captioning-and-captioning-services/correcting-youtube-auto-captions>.
- [10] Benesty, J., Makino, S., & Chen, J. (2005). Speech enhancement. In *Signals and Communication Technology (SCT) series*. Springer.
- [11] Kaladharan, N. (2014). Speech enhancement by spectral subtraction method. In International In *Journal of Computer Applications*, 96(13), 45.
- [12] Wu, C., Li, B., & Zheng, J. (2011). A Speech Enhancement Method Based on Kalman Filtering. In *International Journal of Wireless and Microwave Technologies (IJWMT)*, 1(2), 55.
- [13] Li, Z., Wu, W., Zhang, Q., Ren, H., & Bai, S. (2016). Speech enhancement using magnitude and phase spectrum compensation. In *IEEE/ACIS 15th Institute of Computer and Information Sciences (ICIS)* (pp. 1–4), Okayama, Japan.

- [14] Défossez, A., Usunier, N., Bottou, L., & Bach, F. (2019). Demucs: Deep extractor for music sources with extra unlabeled data remixed. *arXiv preprint arXiv:1909.01174*.
- [15] Li, H., & Yamagishi, J. (2020). Noise tokens: Learning neural noise templates for environment-aware speech enhancement. *arXiv preprint arXiv:2004.04001*.
- [16] Xu, Y., Du, J., Dai, L. R., & Lee, C. H. (2014). A regression approach to speech enhancement based on deep neural networks. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1), 7-19.
- [17] Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.
- [18] Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.
- [19] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in neural information processing systems*, 33, 12449-12460.
- [20] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5206-5210). IEEE.
- [21] Amazon Web Services (AWS). (n.d.). *Amazon Translate API Reference*. Retrieved April 17, 2023, from <https://docs.aws.amazon.com/pdfs/translate/latest/APIReference/translate-api.pdf#welcome>.
- [22] Meta. (n.d.). *About auto-generated subtitles for videos on Facebook*. Meta Business Help Centre. Retrieved April 17, 2023, from <https://www.facebook.com/business/help/593107135335436>.
- [23] Ngo, C., Trinh, T. H., Phan, L., Tran, H., Dang, T., Nguyen, H., Nguyen, M., & Luong, M. T. (2022). MTet: Multi-domain Translation for English and Vietnamese. *arXiv preprint arXiv:2210.05610*.
- [24] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified

- text-to-text transformer. In *The Journal of Machine Learning Research*, 21(1), 5485-5551.
- [25] Youtube. (n.d.). YouTube Data API Overview. *Google Developers*. Retrieved April 17, 2023, from <https://developers.google.com/youtube/v3/getting-started?hl=en>.
- [26] pytube. (n.d.). *pytube/pytube: A lightweight, dependency-free Python library (and command-line utility) for downloading YouTube Videos*. GitHub. Retrieved April 17, 2023, from <https://github.com/pytube/pytube>.
- [27] yashagarwal1411. (n.d.). *yashagarwal1411/SubtitlesForYoutube: Chrome extension for adding external subtitles to a youtube video*. GitHub. Retrieved April 17, 2023, from <https://github.com/yashagarwal1411/SubtitlesForYoutube>.
- [28] Stoller, D., Ewert, S., & Dixon, S. (2018). Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*.
- [29] Arnab. (2021, August 19). Understanding the building blocks of transformers - Analytics Vidhya - Medium. *Analytics Vidhya*. <https://medium.com/analytics-vidhya/understanding-the-building-blocks-of-transformers-c28484788d5a>.
- [30] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, 30.
- [31] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).