AiCE Undergraduate Research Project
Final Report

Fall 2024 Semester 2

Post Call Quality Assurance for 1330 NHSO Contact Center

Team Members

Chavakorn Arunkunarax, Trav , carunku@cmkl.ac.th
Chutikarn Kanchanaart, Ujean, ckancha@cmkl.ac.th
Natcha Soranathavornkul, Baipor, nsorana@cmkl.ac.th
Kasidith Saetang, Copter, ksaetan@cmkl.ac.th

Advisor

Dr. Charnon Pattiyanon

5/5/2568

Table of Contents

Chapter 1

Introduction

1.1 Research Background

The National Health Security Office (NHSO), established in 2002, is a public organization dedicated to ensuring health security for everyone in Thailand under the Universal Health Coverage (UHC) system. With a mission to promote efficient, knowledge-based public good management, NHSO aims to contribute to the nation's sustainability by prioritizing public benefits. It is responsible for developing an accessible and equitable healthcare service system, implementing evidence-based healthcare delivery, providing effective information and communication systems, enabling convenient beneficiary registration, and maintaining robust monitoring and evaluation mechanisms. NHSO's ultimate goal is to ensure that everyone living in Thailand has access to quality healthcare with confidence when needed. [1]

With that goal in mind, the 1330 hotline is established as a key service to ensure accessible healthcare support for everyone in Thailand. This 24-hour helpline offers information, guidance, and assistance related to Universal Health Coverage (UHC), allowing beneficiaries to inquire about healthcare rights, register for services, and report any issues or complaints. The 1330 hotline plays a crucial role in bridging the gap between the public and healthcare providers, ensuring timely and efficient communication to promote confidence and satisfaction in Thailand's health security system.

The National Health Security Office (NHSO) implements a monthly quality assurance process called the Monitoring and Evaluation Protocols to maintain the standard of their services. With the caller's consent, all calls are recorded in two formats: audio and screen recordings. The audio captures the conversation between the agent and the caller, while the screen recording shows the agent's actions during the call, such as searching up information from the knowledge base or filling in the required form into their system called CRM.

The quality assurance process is carried out by two groups: the supervisors and the QA team. There are 269 agents, with 13 QA team members and 20 supervisors at the moment. Each member or supervisor selects four random recordings per each agent to listen to. While doing so, they assess the agent's performance based on specific criteria set by the NHSO.

The evaluation criteria is divided into four main categories: Standard Service, Service Efficiency, Information Recording, and Customer Satisfaction.

1. Standard Service evaluates the agent's overall courtesy and etiquette during the call. This includes their greetings, choice of words, politeness, tone of voice, and more.
2. Service Efficiency focuses on the agent's ability to understand the caller's needs and provide appropriate solutions.
3. Information Recording assesses how accurately and efficiently the agent completes the required forms into their CRM system.
4. Customer Satisfaction is the only category evaluated by the caller themselves. They will be asked to rate their overall experience at6 the end of the call.

1.2 Problem Statement and Significance of the Problem

The current evaluation process requires each supervisor to evaluate 15 agents, four calls per agent, resulting in a total of 60 calls per month. Similarly, each QA team member is responsible for assessing 30 agents under the same protocols, totaling to 120 calls per month. These calls can range from a few minutes to several hours, which leads to a heavy workload on the evaluators. The process is not only time-consuming, but also leaves room for personal bias as fatigue sets in over time from hard work.

This approach can also be unfair to the agents, as the evaluation is based on only four randomly selected calls out of an average of 300 calls handled by one agent. The limited sample size does not provide a comprehensive view of an agent's overall performance. Since the evaluation results directly impact the agents' paychecks, this can lead to concerns about accuracy and fairness.

1.3 Project Solution Approach

To assist the evaluators, the team proposes an automated quality assurance system, where the current process will be streamlined by different technologies and AI models. The voice recording will be used to perform both text and tone analysis, evaluating each recording based on NHSO's criteria introduced in 1.1. The two types of analysis will generate a table with scores assigned to each category of the criteria, offering a clear and structured overview. The final output will replicate the current final reports to ensure convenience for NHSO. Additionally, this project is planned to be developed as a standalone website first. As the team aims to provide a solution that is both helpful and user-friendly, the final product and possible integrations into the NHSO's infrastructure will be further discussed in the near future.

1.4 Project Objectives

1.4.1 Pipeline

Development of a pipeline integrating both text and tone analysis: Conduct research on text and tone analysis technologies, including supporting tools like speech-to-text. Experiment and fine-tune these models to align with NHSO's dataset and evaluation criteria for optimal performance in a short period of time.

1.4.2 Standalone website

Development of a standalone website as proof of concept: Create a platform to demonstrate the streamlined quality assurance process, showcasing the functionality and capabilities of the models in evaluating the recordings.

1.5 Research Question

How can Artificial Intelligence assist the current manual quality assurance process based on the evaluation criteria of the 1330 NHSO Contact Center?

1.6 Research Hypothesis

Integrating text and tone analysis in the same streamlined process should provide comprehensive insights on the voice recordings when evaluated against the criteria. In addition, supporting tools, such as speech to text and speaker diarization, will improve the accuracy of the final result.

1.7 Research variables

1.7.1 Independent Variables:

1.7.1.1 Data Quantity and Quality (i.e. noise, clarity, etc.)
1.7.1.2 Various Models and Techniques
1.7.1.3 Model Hyperparameters

1.7.2 Dependent Variable:

1.7.2.1 The Accuracy and Results Provided by the Model

1.8 Scope of the Research

Throughout the project, we have conducted extensive research on both text and tone analysis. In addition, during this semester, we implemented the finalized evaluation criteria using our synthesized dataset. Furthermore, we completed the development of the processing pipeline and deployed it as a standalone website.

1.9 Limitations of the Research

1.9.1 Lack of Access to Real Data

Due to the privacy concern, we haven't received the dataset from the NHSO yet. However, we have synthesized and used open source data for temporary use. But it is not the actual one, causing numerous limitations in experimenting and researching, especially in data analyzing and augmenting.

1.9.2 Time Constraints

There are a lot of criterias that the QA team and supervisor have to use to assess each agent. Each of them consists of multiple sub criteria. Therefore, with the time given, we were not able to finish every criteria.

1.9.3 Insufficient Criteria Description

The descriptions and examples in each criteria are often insufficient to plan a program or algorithm that covers every case. There were multiple sub criteria that relied on external factors like the current pipeline, from the moment the agent picks up to them holding the call. As well as some that left room for subjective interpretation, this was the main issue as *1.9.5 Communication Challenges* has stated.

1.9.4 Limited Project Scope

This project addresses only one of the four categories in the NHSO evaluation criteria. Moreover, each specific criterion employs its own model and algorithm, as it must independently satisfy its respective requirements.

1.9.5 Communication Challenges

As NHSO is a busy organization, communication between the team and stakeholders often takes a considerable amount of time with frequent miscommunications.

1.10 Research Overview

This research paper is organized into five chapters: Introduction, Background, Methodology, Results, and Conclusion.

The Introduction, which is the current section, outlines the purpose and scope of the research. The Background provides the necessary information, covering technical aspects and a literature review relevant to this study. The Methodology then explains the team's pipeline and technologies used in detail, and outcomes are presented in the Results chapter. Lastly, the Conclusion summarizes the project, highlighting the team's accomplishments, limitations, and lessons learned, along with references cited throughout the paper.

Chapter 2
Background

2.1 Fundamental Technological Theory and Concepts

2.1.1 Speech-To-Text

Speech to text is the process of converting spoken words into a text transcript. It typically combines artificial intelligence-powered speech recognition technology, also known as automatic speech recognition, with transcription. A computer program picks up audio in the form of sound wave vibrations and uses linguistic algorithms to convert the audio input into digital characters, words and phrases. Examples of use cases of speech to text are Call center insight and agent assist, Real-time transcription and translation services, Voice recognition, and etc. [2]

2.1.1.1 Monsoon-Whisper

Monsoon-Whisper is a specialized Thai Speech-To-Text model that builds upon OpenAI's Whisper architecture, optimizing specifically for the Thai language. It was developed by SCB 10X as part of their Typhoon-Audio research initiative. The model is based on the Whisper-Medium and fine tuned on GigaSpeech2 - a massive dataset that contained over 10,000 hours of Thai speech data. The model utilizes a transformer-based encoder-decoder architecture: preprocessing the audio, extracting contextual features from acoustic signals, then finally mapping them to linguistic representations in Thai language. This model may not perform the best, compared to other specialized models, but its strength lies in handling realistic, different audio sources - made specifically to solve the current issue in Thai languages processing. [3]

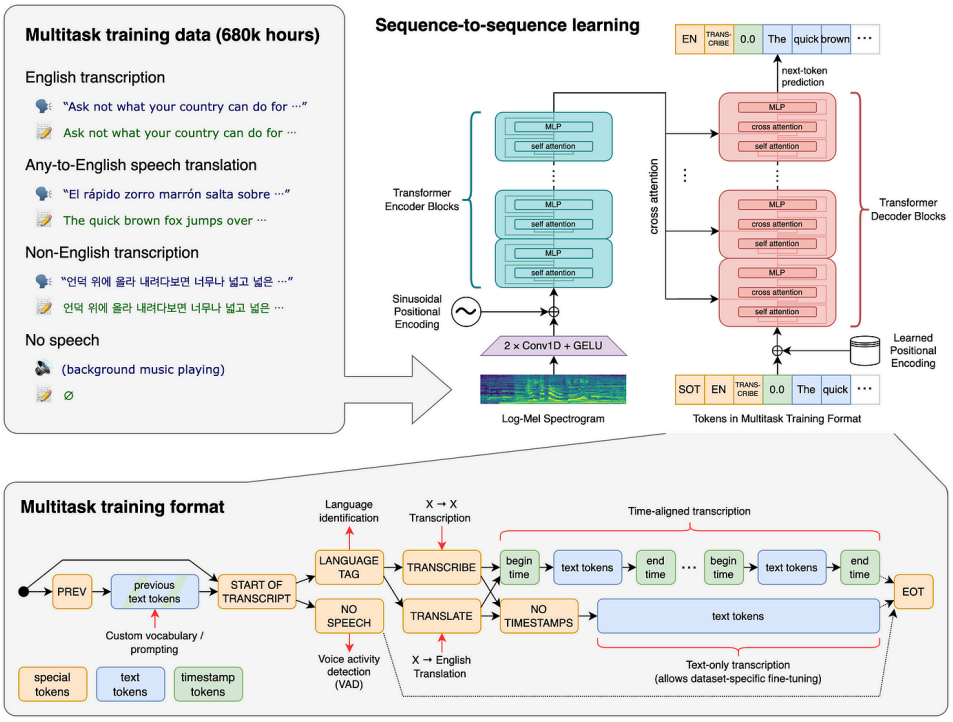


Figure 2.1.1.1.a *Whisper's Structure Diagram (Base Model)*

2.1.1.2 DeepCut

Deepcut is a deep learning-based word segmentation library for Thai. Because Thai does not utilize spaces to separate words, it divides Thai text into words using a bi-directional LSTM (Long Short-Term Memory) neural network. Deepcut can effectively analyze Thai language, even in complicated or confusing circumstances, because it has been pre-trained on vast Thai text datasets. For applications like text analysis, machine learning, and natural language processing (NLP), it offers strong support for tokenizing Thai sentences. Applications like text classification, translation, and sentiment analysis in Thai benefit greatly from the library's ease of use and good Python integration. [4]

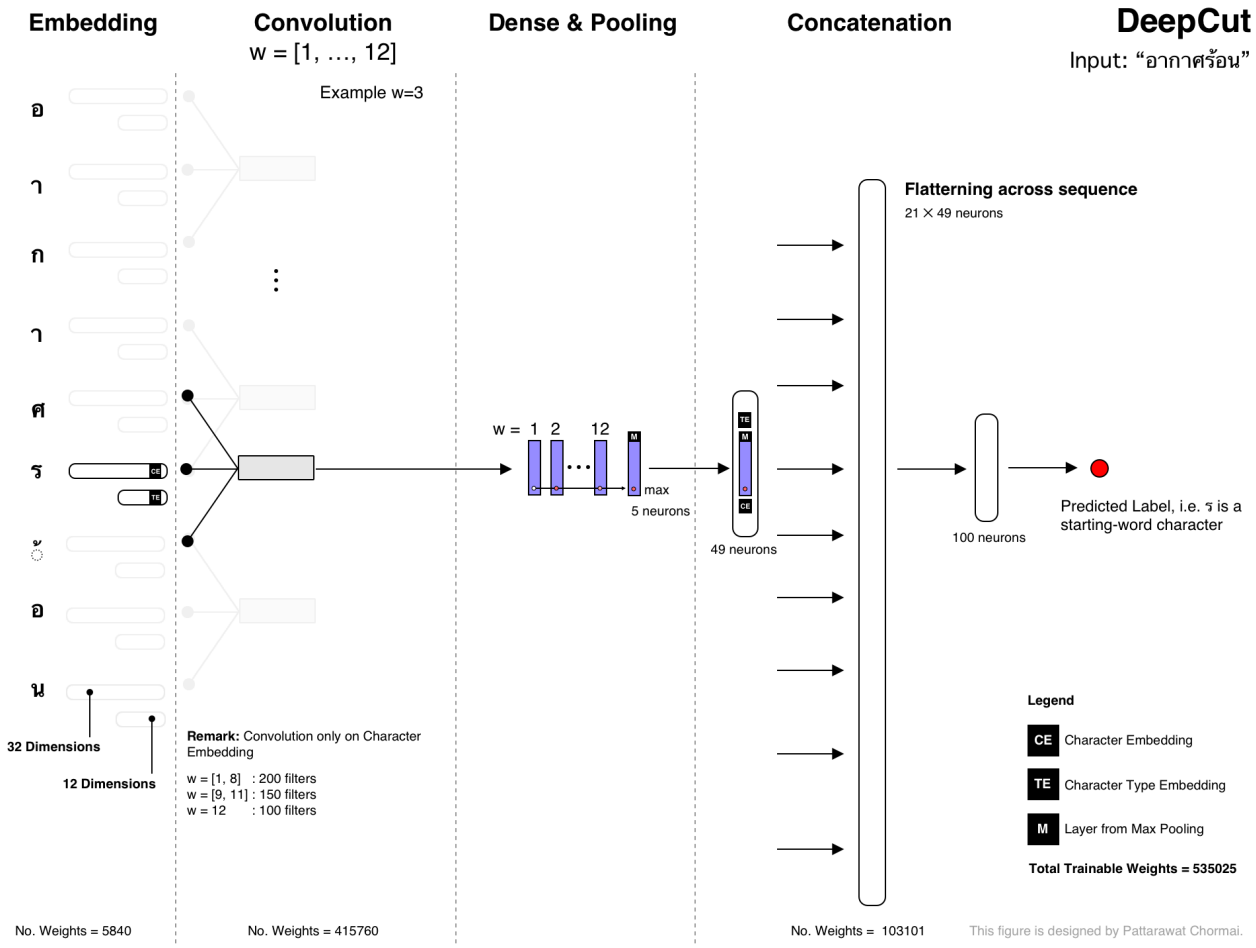


Figure 2.1.1.2.a *DeepCut Structure Diagram.*

2.1.1.3 JiWER

JiWER (Word Error Rate) is a Python module that evaluates the accuracy of speech-to-text systems. It computes the Word Error Rate (WER) by comparing the transcribed text to the ground truth and calculating the number of substitutions, insertions, and deletions needed to align the two.

This measure is a criterion for determining transcription quality. JiWER also enables preprocessing, such as eliminating punctuation or converting text to lowercase, to provide fair and consistent results. Its simplicity and adaptability make it a useful tool for evaluating and developing speech-to-text models. [5]

2.1.2 Speaker Diarization

Speaker diarization segments audio recordings to identify "who spoke when," grouping speech by speaker identity. It's essential for conversational analysis and often used alongside Automatic Speech Recognition (ASR) or Speech Emotion Recognition (SER). Modern methods utilize deep learning for Voice Activity Detection (VAD), audio embedding, and clustering, replacing older statistical approaches. Performance is evaluated with metrics like Diarization Error Rate (DER), Conversational Diarization Error Rate (CDER), and Balanced Error Rate (BER). Ground truth annotations in RTTM format are crucial for accurate evaluation, and tools like Audacity aid in labeling. Speaker diarization enhances ASR and Natural Language Processing (NLP) applications and supports advanced conversational analysis systems. [6]

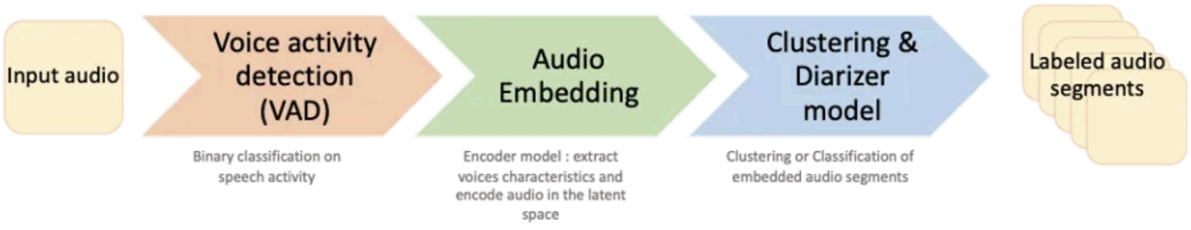


Figure 2.1.2.a *Speaker Diarization Process Diagram*

2.1.2.1 Pyannote-Audio

Pyannote.audio is the python library for speaker diarization.It segments and labels audio into speaker-specific segments using deep learning algorithms. It provides pre-trained models and modular pipelines for tasks like Voice Activity Detection (VAD), speaker embedding, and clustering. Pyannote.audio allows for high-performance audio processing with minimal setup, making it ideal for applications like research, call center analytics, and meeting transcription. [7]

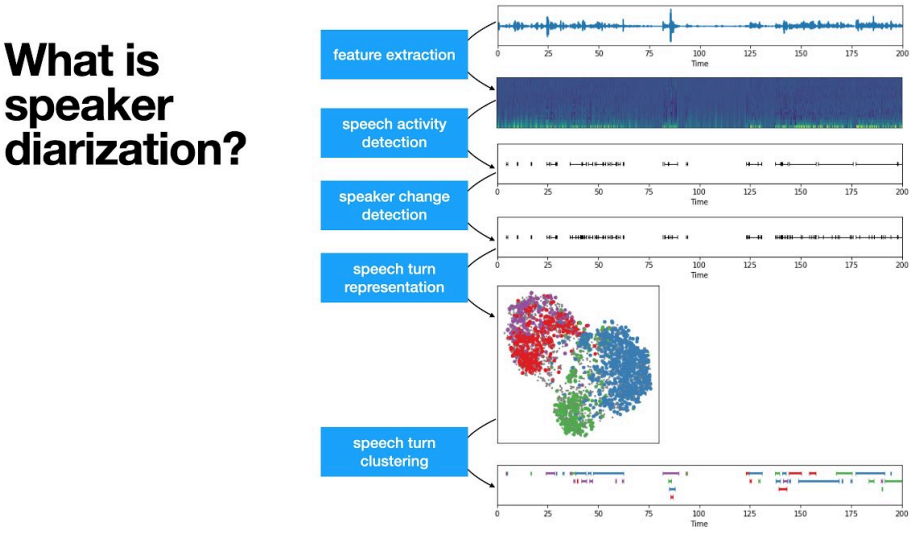


Figure 2.1.2.1.a *Speaker Diarization Process Diagram [8]*

2.1.3 RegEx

RegEx which stands for Regular Expression is a sequence of characters that forms a search pattern. It is commonly used in programming and text processing to find, match, or manipulate strings based on specific patterns.

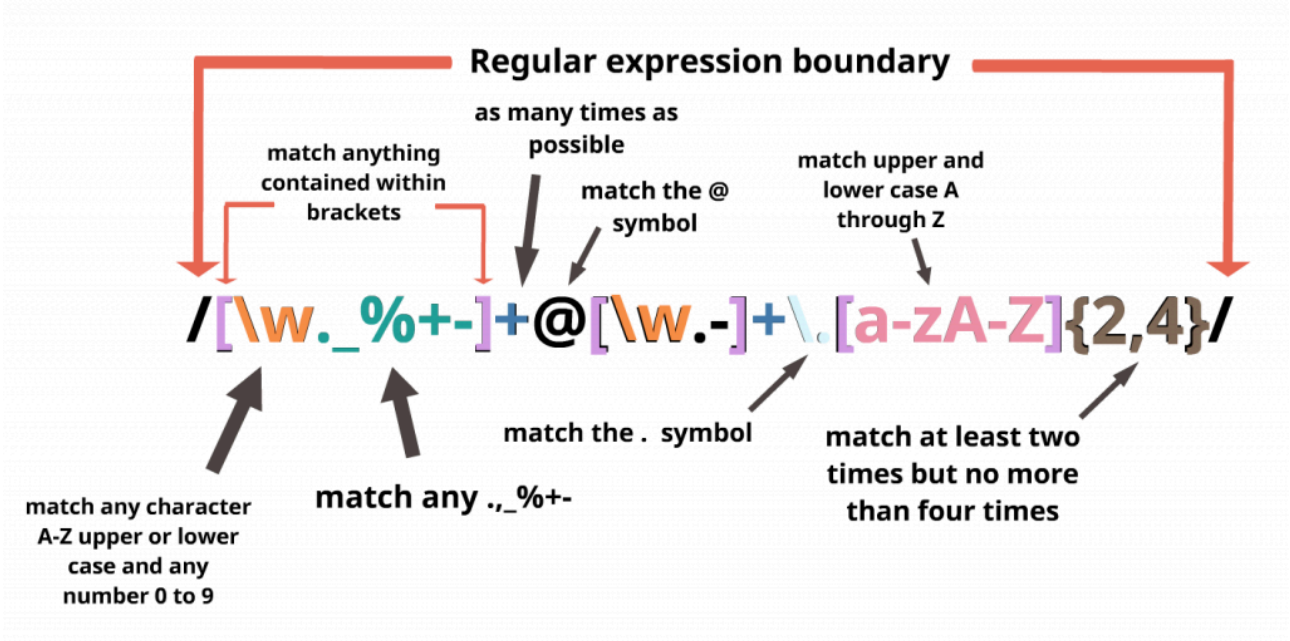


Figure 2.1.3.a *Regular Expression Boundary Of RegEx [9]*

2.1.4 Re's Python Library

The Python Re library is a built-in module in Python used for working with regular expressions. It provides tools to search, match, and manipulate strings based on patterns. [10]

2.1.5 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a subfield of computer science, artificial intelligence, information engineering, and human-computer interaction. This field focuses on how to program computers to process and analyze large amounts of natural language data. [16]

2.1.6 POS Tagging

Part-of-speech (POS) tagging is a fundamental NLP task that involves labeling each word in a text with its corresponding grammatical category, such as noun, verb, adjective, or adverb. The process relies on linguistic rules, statistical models, or machine learning algorithms to analyze word context and syntactic structure. POS tagging is essential for downstream NLP tasks like parsing, named entity recognition, and machine translation. [17]

2.1.7 Text Tokenization

Tokenization is the process of dividing a text into smaller units known as tokens. Tokens are typically words or sub-words in the context of natural language processing. Tokenization is a critical step in many NLP tasks, including text processing, language modelling, and machine translation. The process involves splitting a string, or text into a list of tokens. One can think of tokens as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

2.1.8 WanchanBERTa

WanchanBERTa is a state-of-the-art Thai language model based on the RoBERTa architecture, pretrained on a large corpus of Thai text. It improves upon earlier models by using Byte Pair Encoding (BPE) for subword tokenization, which helps handle the complexities of Thai script and out-of-vocabulary words. WanchanBERTa excels in tasks like text classification, named entity recognition, and question answering, offering better contextual understanding than traditional rule-based or statistical methods. Its pretrained weights and fine-tuning capabilities make it a powerful tool for Thai NLP applications. [18]

2.1.9 PyThaiNLP

PyThaiNLP is a popular Python library designed specifically for Thai natural language processing, offering a wide range of functionalities such as tokenization, POS tagging, named entity recognition, and speech synthesis. It incorporates both rule-based and machine learning approaches to handle Thai language complexities, including word segmentation and spelling correction. PyThaiNLP is widely used in academia and industry due to its ease of integration, open-source nature, and support for modern NLP techniques. It serves as a foundational tool for developing Thai-language AI applications. [19]

2.1.10 Rule-Based NLP

Rule-based NLP relies on predefined linguistic rules and patterns to process and analyze text, rather than using statistical or machine learning models. This approach involves manually crafted grammar rules, dictionaries, and heuristic algorithms to perform tasks like tokenization, POS tagging, and syntactic parsing. While rule-based systems are interpretable and effective for structured languages or specific domains, they lack flexibility and struggle with ambiguity or unseen text patterns. Despite the rise of machine learning, rule-based methods remain useful in hybrid systems, low-resource languages, or applications requiring precise control over linguistic processing.

2.1.11 Acoustic Feature

2.1.11.1 Pitch Features

Pitch features in speech analysis represent the fundamental frequency (F0), where the vocal cords vibrate is perceived as the highness or lowness of voice. Pitch contour tracks these frequencies over time, creating a trajectory that reveals intonation patterns, emotional states, and linguistic information. Pitch dynamics, on the other hand, measures the rate and magnitude of pitch changes, while pitch variation (often quantified as standard deviation) indicates how much a speaker's voice fluctuates from their baseline. These features are particularly important for detecting expressive qualities in speech such as enthusiasm, engagement, and emotional valence. [20]

2.1.11.2 Energy Envelope

Energy envelope represents the amplitude contour of a speech signal over time, capturing how the intensity of speech varies throughout an utterance or round of recording. This feature directly correlates with perceived loudness and stress patterns in speech, making it valuable for detecting emphasis, rhythmic patterns, and overall energy levels in vocal delivery. The energy envelope is typically calculated by measuring the root mean square (RMS) energy in short and overlapping frames of the audio signal, creating a smooth representation of energy fluctuations [21]

2.1.11.3 Mel-Frequency Cepstral Coefficients (MFCCs)

Mel-Frequency Cepstral Coefficients (MFCCs) are spectral features that represent the short-term power spectrum of sound. These features are measured on a nonlinear mel frequency scale, based on a linear cosine transform of a log power spectrum. In short, it mimics the human auditory system's response. MFCCs typically include 12-13 coefficients that capture the vocal tract configuration while filtering out pitch information, making them ideal for representing the timbre and articulation aspects of speech. [22]

2.1.11.4 Spectral Features

Spectral features display the energy distribution across different frequency bands in speech signals, providing insights into the voice quality, articulation, and phonetic content. There are five typical spectral features: centroid, bandwidth, flatness, flux, and tilt.

Spectral centroid represents the "center of mass" of the spectrum. Higher values indicate "brighter" sound (more high-frequency). Bandwidth measures the width of the frequency band that

contains the most energy, indicating spectral spread. Spectral flatness signals how noise-like versus tonal a sound is, with values closer to 1 indicating noise, while values closer to 0 indicating tonal sounds. Spectral flux measures the rate of change in the spectrum over time, capturing articulation dynamics. [23]

Finally, spectral tilt is particularly important for harshness detection as it represents the degree of high-frequency roll-off in the spectrum. It is typically calculated as the ratio between energy in higher versus lower frequency bands. Less steep roll-off (higher spectral tilt) often correlates with perceived harshness or tension in voice. [24]

2.1.11.5 Harmonicity

Harmonicity measures the ratio of harmonic to non-harmonic energy in the signal, providing information about voice quality and the degree of periodicity in speech. Together, these spectral features provide a comprehensive representation of voice timbre and quality that complements temporal and prosodic features in tone analysis [25]

2.1.12 Random Forest

Random Forest is a supervised machine learning algorithm that constructs and combines multiple decision trees to form an ensemble, or "forest," which is used for both classification and regression tasks. Each decision tree in the forest is trained on a random subset of the data and features, introducing diversity and reducing the risk of overfitting. For classification, the final prediction is determined by a majority vote among the trees, while for regression, the average of their outputs is used. This approach leverages the collective intelligence of uncorrelated trees, resulting in more accurate and robust predictions than a single decision tree can provide. [26]

2.1.13 RNN

A Recurrent Neural Network (RNN) is a specialized type of artificial neural network designed to process and analyze sequential data, such as text, speech, or time series, where the order and context of elements are crucial. Unlike traditional feedforward networks, RNNs feature recurrent connections that allow information from previous steps to be carried forward, giving the model a form of memory. This enables RNNs to capture temporal dependencies and patterns within sequences, making them highly effective for tasks like language modeling, speech recognition, and machine translation. However, traditional RNNs face challenges with learning long-range dependencies due to issues like the vanishing gradient problem, which has led to the development of advanced architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) to address these limitations. [27]

2.1.14 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) specifically designed to address the challenges of learning long-term dependencies in sequential data. Unlike standard RNNs, LSTMs use a unique gating mechanism-consisting of input, forget, and output gates-to control the flow of information and maintain a stable memory over long sequences. This architecture enables LSTMs to effectively capture both recent and distant patterns in data, making them particularly valuable for tasks such as natural language processing, speech recognition, and time series prediction. By mitigating issues like the vanishing gradient problem,

LSTMs have become a foundational technology in many modern machine learning applications. [28]

2.1.15 Synthetic Minority Over-sampling Technique (SMOTE)

The Synthetic Minority Over-sampling Technique (SMOTE) is a widely used method for addressing class imbalance in machine learning datasets. Unlike simple duplication of minority class samples, SMOTE generates new, synthetic examples by interpolating between existing minority instances and their nearest neighbors in the feature space. This approach increases the representation of the minority class, helping models learn more general decision boundaries and improving predictive performance, especially for underrepresented groups. By creating more balanced datasets, SMOTE reduces bias toward the majority class and enhances the ability of algorithms to detect rare but important cases. [29]

2.1.16 Individual Learning

Individual learning in machine learning refers to training separate models for each target task or classification objective independently. In the context of tone analysis, this approach involves creating dedicated classification models for specific outcome, with each model optimized specifically for its respective task. Individual learning allows for specialized feature selection, hyperparameter tuning, and threshold optimization tailored to each specific classification problem without interference from other objectives. [30]

2.1.17 Multitask Learning

Multitask Learning (MTL) is a machine learning approach where a single model is trained simultaneously on multiple related tasks, sharing representations between tasks to improve generalization performance. In tone analysis, multitask learning involves training a unified model that simultaneously predicts multiple topics using shared low-level feature representations while maintaining task-specific output layers. This architecture typically includes a shared feature extraction backbone followed by task-specific branches or "heads" that specialize in individual tone criteria. [31]

2.1.18 Paraphrase-multilingual-MiniLM-L12-v2

The paraphrase-multilingual-MiniLM-L12-v2 model is a sentence-transformer designed to convert sentences and paragraphs into 384-dimensional dense vectors, enabling efficient semantic understanding of text. Leveraging a transformer-based architecture, this multilingual model supports over 50 languages and is optimized for tasks such as semantic search, text clustering, and paraphrase identification. Its compact vector representations allow for rapid processing and make it suitable for applications requiring real-time or large-scale analysis. The model’s design balances high accuracy and computational efficiency, making it a valuable tool in modern natural language processing pipelines. [14][15]

2.2 Literature Review

After doing research, there are existing projects that have similar scopes as our project. Some serve the same purpose as a quality assurance tool with artificial intelligence, while others focus on smaller parts like transcription and text summarization. Projects listed below are some of the numerous projects the team has found.

2.2.1 AI QA Tools (Auto scoring tools)

AI auto-scoring tools can automatically evaluate agent performance based on custom criteria and internal quality standards. These tools provide real-time feedback, identify performance trends, and reduce biases typically associated with manual QA evaluations. By transcribing calls and analyzing recorded interactions, they create a comprehensive view of customer engagement. Additionally, these tools reduce the workload of QA teams, enabling more efficient use of resources and allowing staff to focus on higher-value tasks. [11]

2.2.2 Call recording summarizer

While the phone call summaries are available for any user on a website called ‘screenapp’ where they use Ai to summarize the call recording and even a video such as lecture videos. On the website, users can basically upload a call recording file and the system will summarize the call for the user in text. If users choose to summarize the video, they can also simply upload them and the system will summarize the video into texts with many heading topics if available. The website has a lot of features that users can use depending on the user's desire. [12]

Chapter 3
Methodology

This section will be divided into two main segments: Stakeholder's Requirement and Pipeline. They include every process that goes into the experiment design.

3.1 Stakeholder's Requirements

In the initial phase of the project, the team met up with the stakeholders or NHSO to finalize the details and requirements. Our stakeholder wanted an all-in-one interface to evaluate their agents, combining every scoring component together. After discussions, it was concluded that the team will first focus on the first two categories of their current criteria: Standard Service and Service Efficiency.

The team then received the criteria used for evaluation and went through all the conditions to be met in each category, which breaks down into 11 subcategories in Standard Service and 10 subcategories in Service Efficiency. With uncertainties on the dataset, the team focused on 6 criterias from Standard Service: 1.1, 1.3, 1.5, 1.7, 1.9, 1.11, which will be explained in detail in the following paragraphs.

3.2 Pipeline

The pipeline outlines two methods applied to the voice recordings: text analysis and tone analysis. Text analysis focuses on evaluating the context and correctness of word usage by NHSO agents. Meanwhile, tone analysis assesses aspects such as emotion, clarity, intonation, and overall tone. Each analysis evaluates the agents based on different criteria. The results from both methods are then combined to produce a final score. To optimize the pipeline and reduce run time, after speaker diarization is completed, two threads are spawned: one for text analysis and one for tone analysis. This means that after the first step, both analyses are performed simultaneously before the results are combined in the end.

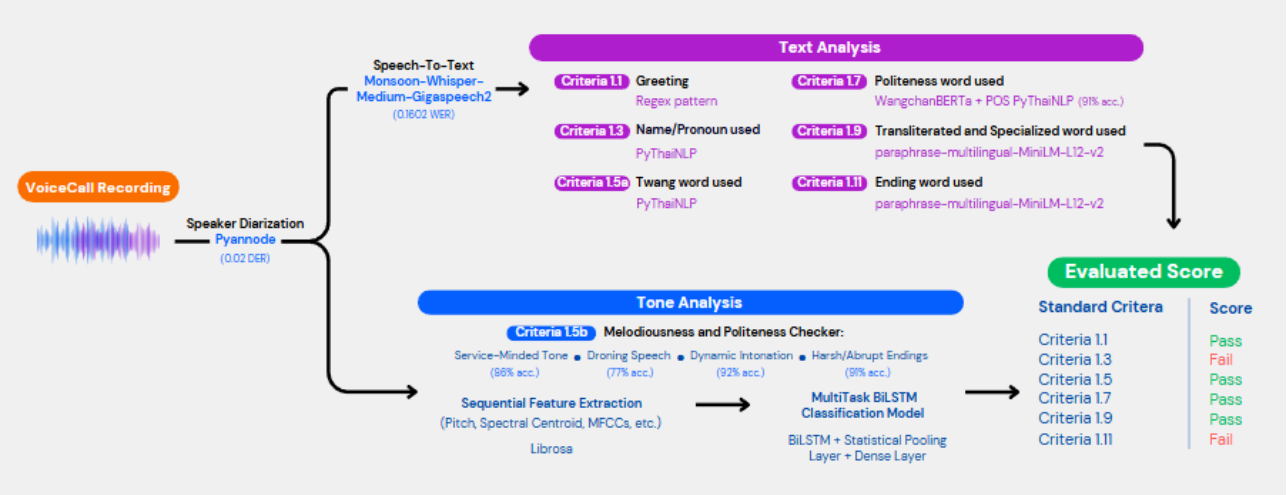


Figure 3.2.a *Pipeline Overview.*

3.2.1 Text Analysis

Since the input is voice call recordings, text analysis cannot be applied directly. The recordings must first be converted into text through a speech-to-text process, with the emphasis on speaker diarization since the scope focuses on the agent's part of the call. Our scope consists of 6 total criterias: 1.1, 1.3, 1.5, 1.7, 1.9, 1.11.

3.2.1.1 Speaker diarization (Pyannote.audio)

Using pyannote.audio, which is the python library for speaker diarization. It segments and labels audio into speaker-specific segments using deep learning algorithms. After inputting the voice recording it will make the timestamp for each speaker. After removing sensitive information from the voice recording, the team went ahead and tried using the audio file as a demo.

3.2.1.2 Speech-To-Text (Monsoon-Whisper-Medium-Gigaspeech2)

Using Whisper-Monsoon which is the Speech-To-Text pre-trained model to do speech to text, it will transcribe the audio to text, dividing audio in small phonetic units (smallest sound in speech) then match the phonemes to the vocabulary words, then match the output with the timestamp through an alignment model. The final output will be the transcript for each speaker with the timestamp.

3.2.1.3 Criteria 1.1

This criterion focuses on verifying the correctness of the agent's greeting line. It ensures that the agent's greeting matches the standard exactly. There are two inputs used for exact matching: the transcript and the regex pattern. Using a Python library called "re," we can apply the regex pattern to search the transcript, resulting in either a match or a non-match.

```
greetings = r"สวัสดีค่ะ.*?รับสายยินดีให้บริการค่ะ | สวัสดีครับ.*?รับสายยินดีให้บริการครับ"
```

Figure 3.2.1.3 *Example of Greeting RegEx Pattern Used*

3.2.1.4 Criteria 1.3

This criteria focuses on the name and pronoun used. Agents should mention the correct name of the citizen and use the appropriate pronoun for each citizen. The stakeholder has given the team a list of approved pronouns and conditions the agents have to follow. Only if the citizen has self-referenced themselves as one of the terms in the given list, can the agent use that pronoun to identify the caller. If the agent had used these specific terms beforehand, that is considered a violation. Apart from these, there were also conditions where the word "พี่" is forbidden in any cases.

3.2.1.4.1 Name Used

The methodology employs a rule-based approach to verify that the agent correctly identifies and formally addresses the citizen by their proper name. To pass this criteria, the agent has to mention the correct name of the citizen. The name could be called wrong at first but should end up being correct after. Moreover, agent must used “คุณ” in front of the name for formal and polite to the citizen. Applying rule-based to track citizen names. Detecting “ชื่อ” remove prefix of the name like นาง นางสาว then collect only citizen first name. This name will be used to evaluate each time the name is mentioned. If the last mentioned name is correct the agent will pass this criteria.

Figure 3.2.1.4.1a *Methodology of Citizen name used using rule-based*

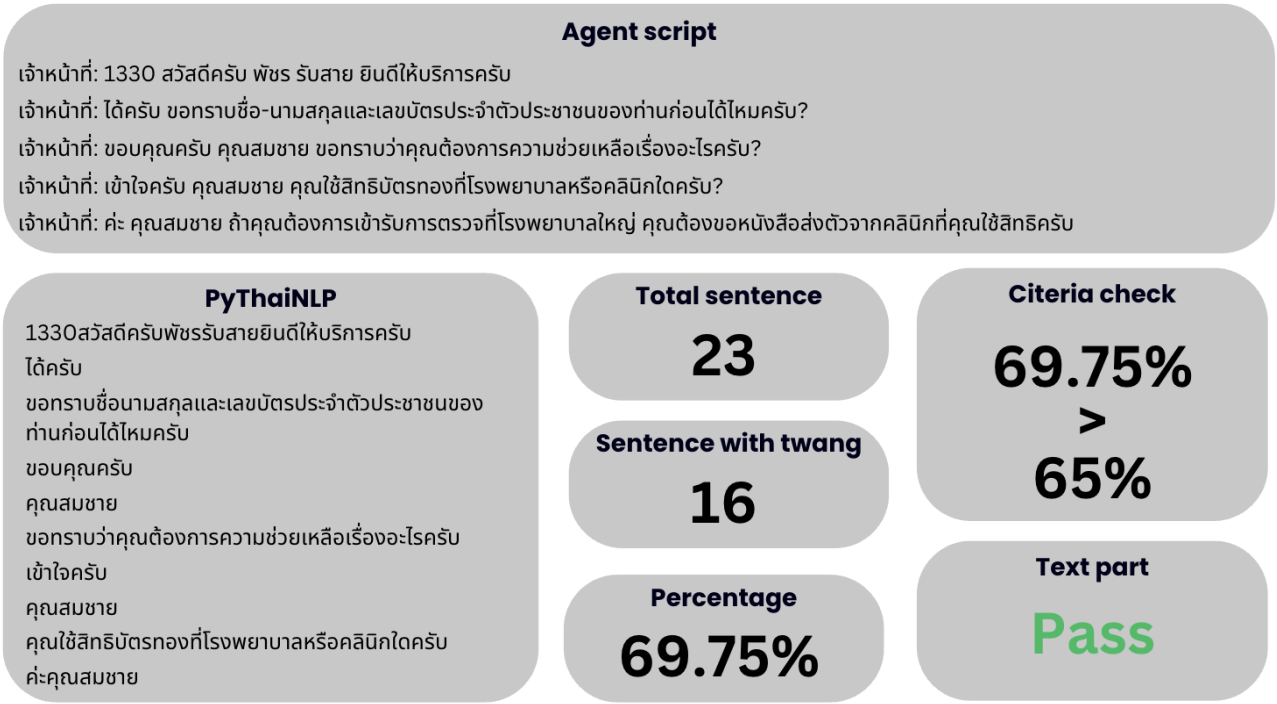
3.2.1.4.2 Pronoun Used

We’ve approached this criteria as a violation detection system, as anything outside the stakeholder’s conditions is marked as correct. To do so, we process the conversation sequentially, alternating between the citizen and agent. The system first tokenizes the conversation using PyThaiNLP’s word segmentation engineer “newmm” and search for the words in the approved pronoun list: family terms (ลุง, ป้า, น้า, อา), monk terms (ท่าน, พระคุณเจ้า), and monk self-reference terms (หลวงพี่, หลวงพ่อ, อาตมา), as well as their prefixed forms with the word “คุณ”. If found, the system validates its position as a pronoun using POS tagging to confirm its part of speech and store it in a list as terms agents are allowed to use. Then, the system processes the agent’s response, where they determine which terms are contextually appropriate based on the previous self-referenced terms. If violations were found, then there is optional comprehensive documentation including the timestamps, line number, and specific violation details for further reports.

Figure 3.2.1.4.2a *Approved Pronoun List, Example, and Methodology of Pronoun Detection*

3.2.1.5 Criteria 1.5 (Text Analysis)

We evaluate the agent's use of twang words to ensure a culturally appropriate and engaging tone. The process involves filtering out the citizen's speech to isolate the agent's speech. Using the PyThaiNLP library, the agent's speech is segmented into sentences. We count the total number of sentences and the number of sentences containing twang words (ครับ คับ คะ ค่ะ). We then find the percentage of sentences containing twang words compared to the total sentence before comparing with the passing percentage for this part.



3.2.1.6 Criteria 1.7

This criterion focuses on appropriate words used, including bad words detection, negative content detection and lastly the specific word used in specific part of speech which is our stakeholder requirement.

To satisfy this criterion, agents must meet two distinct language requirements. First, the system conducts bad word and negative content detection by analyzing the agent's speech transcripts. Using WangchanBERTa, a Thai pre-trained model with 91% accuracy in Thai text classification, each line of the agent's speech is evaluated for inappropriate language. If any offensive terms or negative content are detected, the agent automatically fails this criterion, ensuring all communications maintain professional standards.

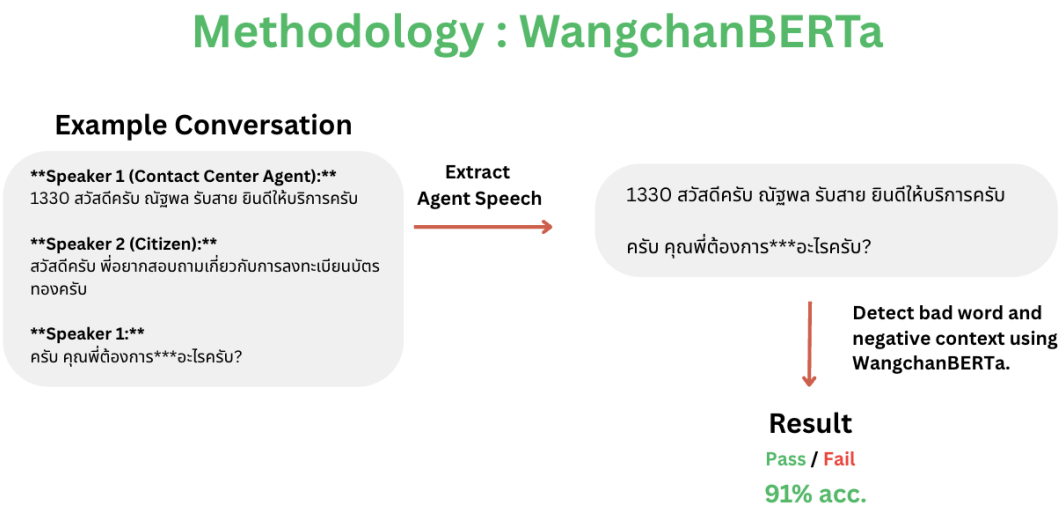


Figure 3.2.1.6a *Methodology of Bad word & Negative Content Detection*

Second, the system enforces proper usage of specific words through part-of-speech (POS) analysis. Focusing on the words "มัน" and "เอา," the evaluation employs PyThaiNLP's text tokenization and POS tagging capabilities to verify these terms are not used in grammatical contexts prohibited by stakeholder requirements in order to enhance level of the thai language. Agents must comply with both requirements to successfully pass this criterion.

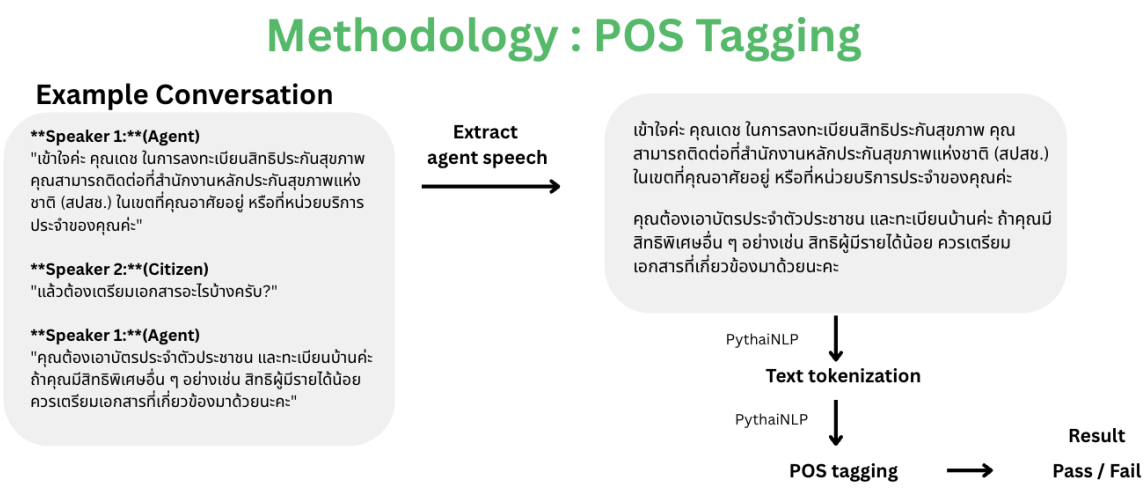


Figure 3.2.1.6b *Methodology of POS Tagging*

3.2.1.7 Criteria 1.9

This criterion focuses on transliterated and specialized words. For transliterated words, it ensures that the agent does not mention a transliterated word before the citizen does. The method involves first creating a list of transliterated words. Then, both the citizen's and agent's lines are checked. If the citizen uses any word from the list, it is removed. On the other hand, if the agent mentions a word from the list before the citizen does, the agent will immediately fail the criterion.

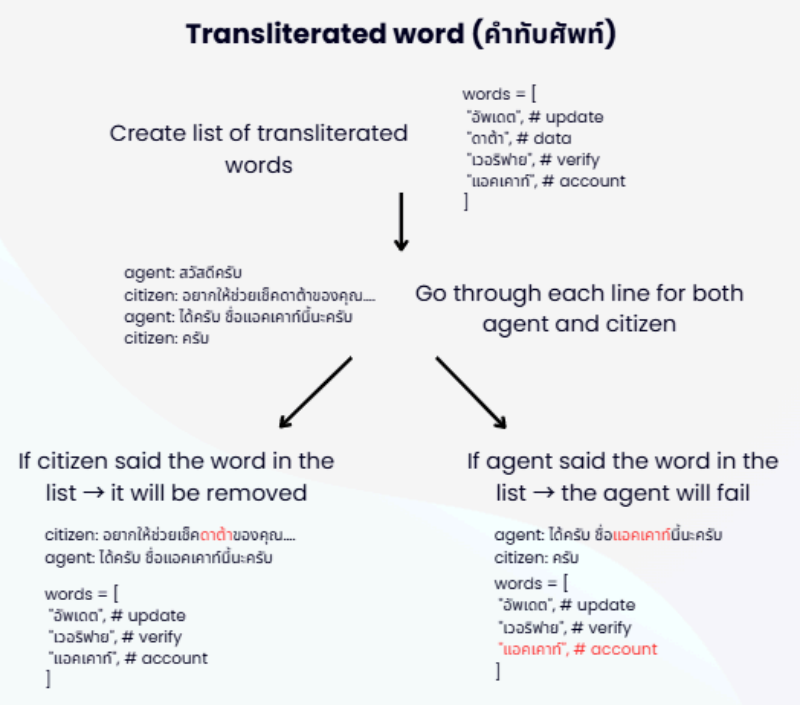


Figure 3.2.1.7a *Methodology of Transliterated Word*

For specialized words, the criterion states that if the agent uses a specialized word, they must provide its meaning afterward. To implement this, I created a dictionary containing specialized words and their corresponding meanings. Then, I analyzed only the lines spoken by the agent. If the agent uses any of the specialized words, the system checks the sentence containing the word and the

two sentences that follow. It then calculates the cosine similarity between the predefined meaning and these sentences. If the similarity exceeds the threshold, it is considered a pass; otherwise, it results in a fail. To calculate the similarity, I used the model *paraphrase-multilingual-MiniLM-L12-v2*, which is well-suited for measuring sentence similarity.

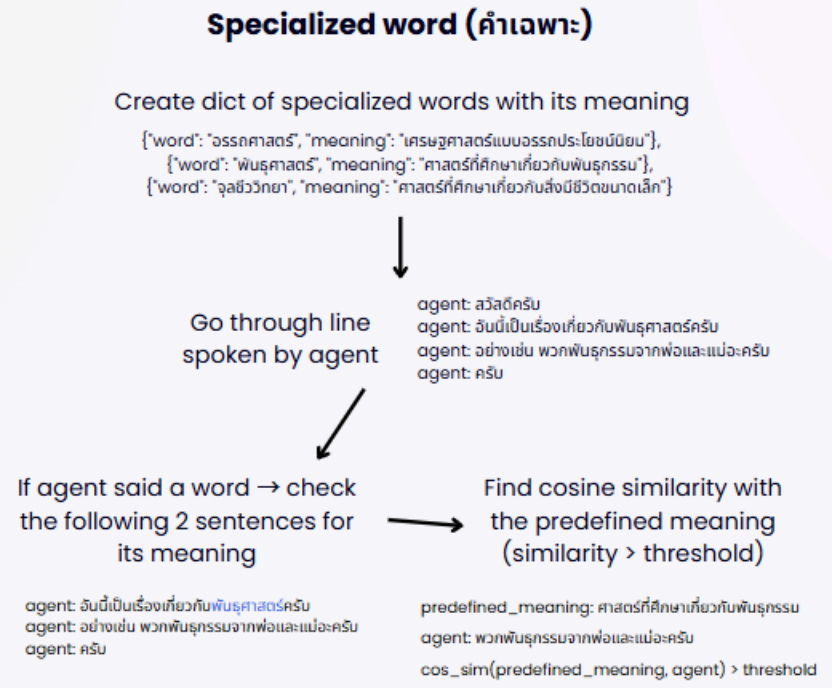


Figure 3.2.1.7b *Methodology of Specialized Word*

3.2.1.8 Criteria 1.11

This criterion focuses on the agent's ending phrase. While there is no exact required wording, the phrase must meet the standard set by the stakeholder. To evaluate this, I created an ideal ending phrase and compared it to the agent's lines. If the cosine similarity between an agent's line and the ideal ending phrase exceeds the threshold, it is considered a pass; otherwise, it results in a fail. To calculate the similarity, I used the model *paraphrase-multilingual-MiniLM-L12-v2*, which is well-suited for measuring sentence similarity.

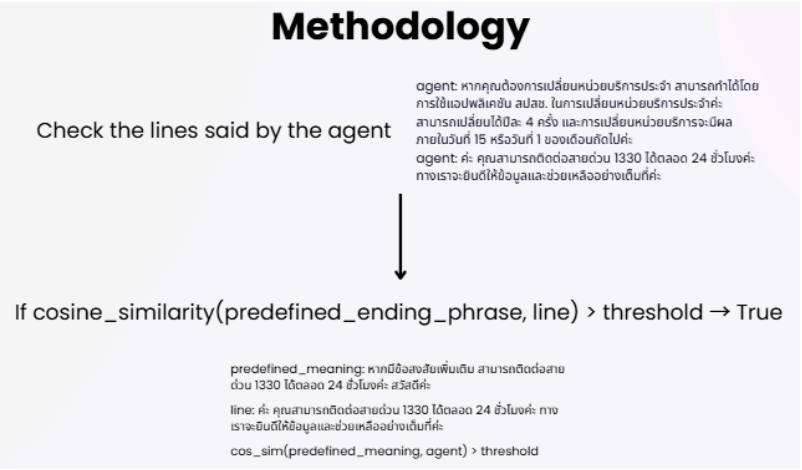


Figure 3.2.1.7b *Methodology of Criteria 1.11*

3.2.2 Tone Analysis (Continuation of Criteria 1.5)

Tone analysis through acoustic features in call recordings can provide valuable insights into contact center agent behavior by revealing their emotional states and communication effectiveness. These evaluate how the agent speaks, thus, is better suited for tone analysis compared to text. By selecting and analyzing specific features of the agent's recordings, we ensure both the validity of our analysis and interpretability of the result - the key factor that influenced the entire approach. There are 4 main criterias to evaluate:

1. Service-Minded Tone (น้ำเสียงเต็มใจให้บริการ):
    a. Whether the agent's tone conveys willingness and enthusiasm to provide service.

2. Non-Sluggish Speech (ไม่เนือย):
    a. Whether the agent maintains appropriate energy and pace

3. Dynamic Intonation (ไม่ราบเรียบ):
    a. Whether speech contains sufficient pitch variation rather than being monotonous or flat

4. No Harsh/Abrupt Endings (ไม่ห้วน):
    a. Whether speech maintains appropriate vocal quality without harsh, abrupt, or excessively tense delivery, particularly at phrase endings.

3.2.2.1 Phase One - Data Acquisition and Preprocess

With the lack of the real dataset, we first had to create a reliable dataset and label. In the first phase, we focus on data acquisition and preprocessing for the final implementation. We've opted for an open source one from ThaiSER (Thai Speech Emotion Recognition) since emotion and tone are correlated to one another. The initial dataset contained 27,854 recordings totaling 41 hours and 36 minutes, each with assigned emotion labels. After reviewing ThaiSER's data collection process and sampling the dataset, we established specific standards to best simulate the real scenario: only improvisational recordings (not scripted ones which sounded forced), majority-vote emotion labels from at least 5 people per recording, recordings longer than 5 seconds to ensure valuable information, and only recordings from 360-degree Lavalier microphones that most closely resemble real-life situations. After preprocessing the dataset to these standards, the refined dataset included approximately 14,000 recordings distributed across five emotions: Angry (3,111), Frustrated (7,864), Happy (4,070), Neutral (7,359), and Sad (2,781).

3.2.2.1.1 Feature Engineering

For interpretability, we selected and analyzed fundamental features so the results can be traced back and explained. We extracted 14 static acoustic features from the audio recordings to support the initial label generation process.

These features include
- Pitch statistics: standard deviation, mean, and jitter
- Energy statistics: mean and standard deviation
- Spectral centroid: the "center of mass" of the frequency spectrum
- Spectral bandwidth: the width of the frequency range containing most energy
- Spectral flatness: measuring how noise-like versus tonal the sound is

- Two MFCC coefficients: mfcc_1 and mfcc_2
- Harmonicity and Harmonic-to-Noise Ratio (HNR)
- Speech rate
- Trailing slope: the pitch movement at the end of utterances (later removed after data analysis revealed it contained only zero values.)

These static features provided a comprehensive numerical representation of each recording's acoustic properties, allowing me to make data-driven adjustments to the emotion-based labels.

3.2.2.1.2 Random Forest and Feature Importance Analysis

From *feature engineering*, we use the static acoustic features as input paired with emotion labels to train four separate random forest models. With no existing classification labels for each tone criterion, we needed to generate reliable training labels. We began by creating initial semi-supervised labels through a combination of emotion-based mapping and acoustic features. For the emotion-based component, we mapped emotion categories to tone criteria using probability distributions (i.e. 80% of happy recordings marked as "Pass" for Service-Minded Tone, while only 10% of angry recordings would pass this criterion). We then made small adjustments based on acoustic features using predefined experimental weights. The Random Forest models provided quantitative importance scores for each acoustic feature relative to each tone criterion, revealing key relationships such as pitch variation being highly predictive for non-monotone classification, while energy features were more important for non-sluggish detection. This allowed us to analyze feature importance and understand which acoustic features were most predictive for each criterion.

3.2.2.1.3 Final Datasets and Labels

Then from the feature importance analysis, we created improved importance-based final labels by replacing the initially predefined weights with the data-driven feature importance weights from the Random Forest models. We adjusted the weighting formula to rely 60% on emotion-based mapping (maintaining its role as the primary signal since these labels were reliable and valid) and 40% on the acoustic features weighted by their importance scores. We also introduced small random noise to the calculations to simulate real-life data variability and prevent straightforward mathematical patterns in the labels. The result was a set of binary labels for each of the four criteria (Service-Minded Tone, non-sluggish speech, Dynamic Intonation, No Harsh/Abrupt Endings) across the approximately 14,000 recordings, resulting in four different datasets, one for each criteria.

3.2.2.2 Phase Two - Model Implementation

Now that we have our final dataset with labels, we moved forward into training our models. We implemented two different approaches to compare their effectiveness: individual models for each tone criterion and a multi-task LSTM model capable of predicting all criteria simultaneously.

3.2.2.2.1 Feature Extraction

In phase two, we transitioned from static feature extraction to sequential extraction as they provide more information overall. They track how acoustic properties change throughout the speech, providing a better representation.

First, we extracted the pitch contour (F0) using the PYIN algorithm, which provides fundamental frequency estimates frame by frame, along with a voiced flag indicating whether each frame contains voiced speech. We used frame and hop lengths of 1024 and 256 samples respectively, ensuring fine-grained temporal resolution.

We calculated the energy envelope using the root mean square (RMS) energy in each frame, capturing how speech intensity varies over time. For spectral information, we extracted 13 MFCCs along with their delta (first derivative) and delta-delta (second derivative) coefficients, which represent how the vocal tract configuration changes over time. We added several spectral time-series features: spectral centroid, spectral bandwidth, spectral flatness, and spectral flux.

Beyond these standard features, we engineered criteria-specific features targeted at our specific tone analysis needs. For "Service-Minded Tone," we computed pitch dynamics as the gradient of the pitch contour, highlighting rapid pitch changes associated with engaged speech. For "Non-sluggish speech," we calculated onset strength to detect speech events or syllables, providing information about speech rhythm and rate. For "Dynamic Intonation," we created specialized features including scaled pitch variation and pitch range. For "No Harsh/Abrupt Endings," we computed spectral tilt as the ratio of high-frequency to low-frequency energy across time, which correlates with perceived voice harshness.

After extraction, we normalized all features using pre-trained scalers saved from the training phase. We then addressed the variable-length nature of our recordings by padding shorter sequences or truncating longer ones to a fixed length of 500 time steps. This standardization was essential for batch processing in our LSTM models. Finally, we prepared the features for model input by stacking them appropriately and ensuring the correct dimensionality for the LSTM layers, resulting in a tensor of shape [batch_size, time_steps, features].

This sequential feature extraction approach allowed our models to analyze the temporal dynamics of speech, capturing prosodic patterns that static features alone could not represent.

3.2.2.2.2 Individual Models

For our individual models approach, we implemented separate LSTM-based neural networks for each of the four tone criteria (Service-Minded Tone, non-sluggish speech, Dynamic Intonation, and No Harsh/Abrupt Endings). Each model shared the same foundational architecture but was trained independently using only the relevant criterion's labels.

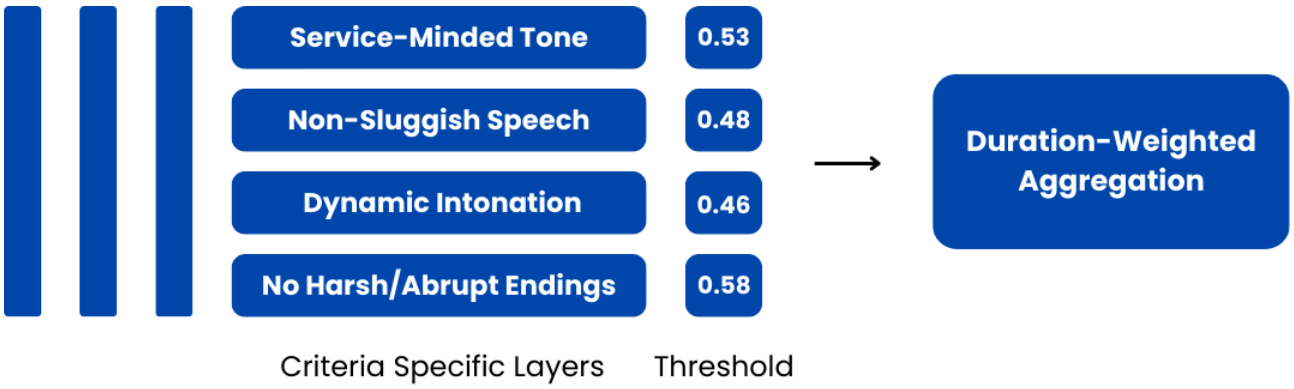
The architecture of each individual model began with an input layer accepting sequential features of variable length with our feature dimension. This was followed by a base sequential model component consisting of two stacked bidirectional LSTM layers, each with 128 units. Bidirectional processing allowed the network to analyze the speech signal in both forward and backward directions, capturing contextual information from both past and future frames - essential for prosodic analysis where tone patterns may span in both directions.

We applied dropout regularization (rate=0.3) between LSTM layers to prevent overfitting. The output from the bidirectional layers then passed through a custom statistical pooling layer, which computed the mean, standard deviation, and maximum values across the time dimension. This pooling approach effectively captured both the central tendency and variability of features across time, transforming variable-length sequences into fixed-dimension representations.

Following the pooling layer, we added dense layers of decreasing size (128, then 64 neurons) with ReLU activation functions, each followed by dropout for regularization. The output layer consisted of a single neuron with sigmoid activation to produce a probability between 0 and 1, representing the likelihood of the recording passing the specific tone criterion.

Each individual model was trained independently using binary cross-entropy loss and the Adam optimizer with an initial learning rate of 0.001. We implemented early stopping with a patience of 10 epochs to avoid overfitting. To address class imbalance in our training data, we applied the SMOTE technique to generate synthetic samples of the minority class for each criterion separately.

3.2.2.2.3 Multi-Task LSTM



3.2.2.2.a Multi-Task LSTM Model Architecture

To experiment, we also tried two multi-task models (one with SMOTE applied per criteria and one without) as all the acoustic features are connected to one another. Having one shared based layer would theoretically allow the model to become more accurate after discovering and learning new correlations.

These models shared the same architecture, the same base sequential model as our individual models - stacked bidirectional LSTMs with 128 units and statistical pooling - but differed in how the shared representation was used.

After the statistical pooling layer, we implemented a shared dense layer of 256 neurons with ReLU activation to capture common patterns relevant to all tone criteria. From this shared representation, we created four separate branches (one for each criterion), each consisting of a 64-neuron dense layer with ReLU activation and dropout, followed by a single-neuron output layer with sigmoid activation. This structure allowed the model to share information across tasks while maintaining criterion-specific processing in the final layers.

The multi-task model was trained with a combined loss function that aggregated the binary cross-entropy losses from all four outputs, with each criterion contributing equally to the total loss. We used the Adam optimizer with a slightly lower learning rate (0.001) and early stopping based on the validation loss.

To address class imbalance in the multi-task context, we implemented a specialized version of SMOTE specifically designed for multi-task learning. This approach created synthetic samples

that balanced all criteria simultaneously. Unlike the independent SMOTE applied to individual models, this approach considered the joint distribution of all four criteria labels, ensuring that all combinations of criteria were adequately represented in the training data.

For both approaches, we implemented threshold optimization on the validation set to maximize F1 scores. Rather than using the standard threshold of 0.5, we found the optimal decision threshold for each criterion that maximized the F1 score on the validation data. These optimized thresholds were then stored in a json file and used during inference in the deployment pipeline.

Chapter 4
Results

4.1 Speaker Diarization (PyAnnote)

The result ranges from 1.7% to 47%. The model gains a high accuracy if the speakers are different genders.

Speakers	DER
Male and Female	1.72%
Male and Male	30.24%
Female and Female	46.73%
Total	10.63%

Figure 4.1 *Diarization Error Rate of the PyAnnote Model*

4.2 Speech-To-Text Results (Whisper Monsoon)

The average accuracy of the speech-to-text model after fine-tuning is around 16%, which is lower than that of the open-source model.

Speaker SPEAKER_00 (135.9s - 141.9s)
สิทธิ์ รับประกัน สุขภาพแห่งชาติ หรือ สิทธิ์ บัตรทอง จะ คุ้มครอง ใน การรักษา ทุกโรค ตาม ข้อบ่งชี้ ทางการแพทย์ ครับ

Speaker SPEAKER_00 (142.2s - 150.3s)
แพทย์ประเมิน ว่า คุณ มีความจำเป็น จะ ต้อง ได้รับ การ ตรวจ หรือ การรักษา วิธี ไหน สามารถ ใช้สิทธิ์รักษา ได้ ฟรี ไม่ เสียค่าใช้จ่าย ครับ

Speaker SPEAKER_00 (150.8s - 157.0s)
ในเรื่องของ การ ตรวจ สุขภาพประจำปี สิทธิ บัตรทอง แต่ ไม่ ได้ มี บริการ ตรวจ สุขภาพประจำปี นะครับ

Speaker SPEAKER_01 (158.8s - 167.5s)
อืมเรา ผม สามารถ เช็ค สิทธิ์ โรงพยาบาล ที่อยู่ ใน ที่ เข้าร่วม ได้ ที่ไหน ปั๊ม ไหม ครับ

Speaker SPEAKER_00 (171.2s - 173.5s)
หมายถึง หมายถึง พูดต้องการ จะ ไป ทำ อะไร ครับ

Speaker SPEAKER_01 (174.9s - 181.5s)
คือถ้าผมอยากรู้ ว่า โรงพยาบาล ไหน ที่ เข้าร่วม ศิษย์ บ้าง นะ ครับ ผม ต้อง เช็ค ใน ทาง ไหน บ้าง ครับ

Speaker SPEAKER_00 (184.7s - 194.5s)
ผม ถ้า คุณ ต้องการ ตรวจสอบ เครือข่าย หน่วยบริการ ใน ระบบ สปสช สามารถ ที่จะ ตรวจสอบ ผ่าน ทาง

เว็บไซต์ สปสช ก็ได้ ครับ หรือ หาก คุณ ต้องการ ที่จะ ย้าย สิทธิ์ การรักษา

Speaker SPEAKER_00 (194.9s - 207.5s)

คุณ สามารถ ดำเนินการ แจ้งย้าย ผ่าน แอปพลิเคชัน สปสช หรือ ทำรายการผ่าน ไลน์ สปสช ก็ได้ ครับ ในขั้นตอน สุดท้าย นะ ครับ ที่ ให้ เรื่อง หน่วยบริการ ก็คือ จะ มี การ ขึ้น โชว์ หน่วยบริการ ใน ระบบ ไม่ เลือก ครับ

Speaker SPEAKER_00 (207.7s - 210.6s)

ประจักษ์ โชว์ เลย ว่า คุณ สามารถ เลือกหัว บริการ ไหน ได้ บ้าง

Speaker SPEAKER_00 (211.6s - 216.0s)

แต่จะ ยึดตาม พื้นที่พักอาศัย จริง นะ ครับ เช่น หาก พักอาศัย อยู่ ใน เกล็ด พื้นที่

Speaker SPEAKER_00 (216.8s - 224.8s)

ลิ่งชันทร์ ตามทะเบียนบ้าน คุณ ว่า จะ สามารถ เลือก หน่วยบริการ ตาม เขตพื้นที่ ตลิ่งชั้น ที่ มี การ เปิด ณ ตอนนี้ ได้ เท่านั้น ไม่ สามารถ เลือก ข้ามเขต ได้

Speaker SPEAKER_01 (226.2s - 227.1s)

อืม

Audio	Distilled Whisper TH	Monsoon	FIne Tuned Monsoon
Medical Audio1	0.3822	0.3337	0.3222
Medical Audio2	0.3005	0.2989	0.3217
Medical Audio3	0.2488	0.283	0.2146
Medical Audio4	0.3157	0.3337	0.3333
TestAudio1	0.1299	0.0909	0.1299
TestAudio2	0.2222	0.1852	0.2037
TestAudio3	0.1447	0.1447	0.0263
TestAudio4	0.1818	0.0364	0.0
TestAudio5	0.3611	0.0972	0.25
TestAudio6	0.1047	0.1395	0.0465
TestAudio7	0.1042	0.1042	0.0625
TestAudio8	0.0678	0.2373	0.0847
TestAudio9	0.0676	0.1081	0.0676
TestAudio10	0.122	0.0854	0.061
v2TestAudio1	0.2824	0.3294	0.3294
v2TestAudio2	0.2613	0.1532	0.1441
v2TestAudio3	0.1471	0.1275	0.1373
v2TestAudio4	0.4478	0.194	0.2687
v2TestAudio5	0.4074	0.6852	0.2778
Average	0.2274	0.2033	0.1602

Figure 4.2 *Word Error Rate of the STT model*

4.3 Criteria 1.1

The result of exact wording depends on the speech-to-text model because this process is a simple coding algorithm that uses Python's Re library to search for RegEx patterns in the transcript. The result will either be a match or a non-match, depending on whether the transcript contains the specified RegEx pattern. In our experiment, the program successfully found the regex pattern wanted, specifically the greeting phrase.

4.4 Criteria 1.3

This criteria is admittedly underdeveloped compared to the rest, with both technical constraints and insufficient information on the criteria. For these reasons, we've decided to divide the result up into the sub-criteria accordingly (delete dai)

When testing the algorithm with our synthesized text data, the system achieved an accuracy of approximately 35%, successfully detecting 3,133 names out of 8,960 conversations. However, since the system relies on rule-based methods, it has inherent limitations. Specifically, if a user does not include the keyword "ชื่อ" (name) before their first name, the system fails to identify the name correctly.

As for pronouns, testing the algorithm with our synthesized text data resulted in a 87.5% true positive and 19.1% false positive, where we had 8960 conversations that are marked as pass and 2560 conversations as failed. The program was able to detect only the typical test cases, such as detecting premature self-referenced by the agents with all of the given terms and detecting the forbidden terms. For these cases, the program is highly effective and it is a combination of rule based and NLP techniques. However, advanced test cases like detecting changes in self-referenced or identifying the correct pronoun to use when there are possessive and compound nouns remains to be discussed with the stakeholders.

To validate these results, we have printed out the result in a JSON file with violations made. It is safe to say that the program is effective, however, the numbers aren't accurate. This is because there were several conversations, when evaluated separately per sub criteria, that were mistakenly labeled. There were multiple violations found in the *pass* dataset that were correct after manual evaluation and vice versa. This criteria remained to be developed and validated with the stakeholder's data and further explanation.

4.5 Criteria 1.5

The scoring process combines the tone analysis results from both voice and text components. After obtaining the tone score from the voice analysis and the text analysis, these scores are summed and divided by 5 to normalize the result against the established threshold. A score exceeding 0.6 (equivalent to 3/5) is considered a passing mark for this criteria, indicating that the agent's tone meets the required standards for cultural appropriateness and engagement.

4.5.1 Text Analysis

The percentage of sentences with twang words relative to the total is calculated, with a threshold of 65% set as the passing percentage for text part, based on synthesized data where percentages below 65% were deemed inappropriate. We find numerous way to detect the politeness through text include:

- Number of twang words per total sentences: 60% to 111% range.
- Sentences-ending with twang words per total sentences: 36% to 89% range.
- Sentences with twang words per total sentences: 52% to 89% range.

The third method has the narrowest range and thus easier to find the passing percentage.

4.5.2 Tone Analysis

Since we've experimented with 4 different models in total, this section will include results of them all.

4.5.2.1 Random Forest

น๊าเสียงเต็มใจให้บริการ:	**ไม่เนือย:**	**ไม่แข็ง:**	**ไม่ราบเรียบ:**
accuracy: 0.6223	accuracy: 0.4809	accuracy: 0.4373	accuracy: 0.7017
precision: 0.2626	precision: 0.3604	precision: 0.4322	precision: 0.6503
recall: 0.6872	recall: 0.8578	recall: 0.9935	recall: 0.6489
f1: 0.3800	f1: 0.5076	f1: 0.6024	f1: 0.6496

The Random Forest results reveal varying performance across the four tone criteria, with accuracy ranging from 43.73% to 70.17%. This approach shows several characteristic patterns worth noting. The models generally demonstrate high recall but lower precision, suggesting they tend to classify many samples as positive but with a handful of false positives. This imbalance is particularly evident in the "**ไม่แข็ง**" (No Harsh/Abrupt Endings) criterion, which has an exceptionally high recall (99.35%) but low precision (43.22%) and accuracy (43.73%). The "**ไม่ราบเรียบ**" (Dynamic Intonation) criterion achieves the most balanced performance with comparable precision and recall values, resulting in the highest F1 score (64.96%). These results reflect the challenges using static feature-based classification on prosodic qualities, where temporal patterns are significant but not fully captured by whole statistics.

4.5.2.2 Individual, Multitask, Multitask with Separate SMOTE

	Individual Models	Multitask Model	Multitask Model with Separate SMOTE
Service-Minded Tone	0.84 acc. F1= 0.77 AUC = 0.82	0.86 acc. F1= 0.79 AUC = 0.83	0.87 acc. F1= 0.80 AUC = 0.85
Non-Sluggish Speech	0.76 acc. F1= 0.84 AUC = 0.88	0.78 acc. F1= 0.84 AUC = 0.87	0.78 acc. F1= 0.85 AUC = 0.89
Dynamic Intonation	0.90 acc. F1= 0.94 AUC = 0.96	0.91 acc. F1= 0.93 AUC = 0.95	0.93 acc. F1= 0.95 AUC = 0.97
No Harsh/Abrupt Endings	0.91 acc. F1= 0.91 AUC = 0.93	0.91 acc. F1= 0.90 AUC = 0.92	0.91 acc. F1= 0.91 AUC = 0.94
		86.5% avg	87.8% avg

Figure 4.5.2.2.a Accuracy and Evaluation Metrics Diagram

For the Individual models approach, our LSTM networks demonstrate strong performance across all four tone criteria, with accuracies ranging from 76% to 91%. Each model benefits from dedicated focus on a single classification task, allowing specialized tuning of hyperparameters and thresholds. The "Dynamic Intonation" (Not monotone) and "No Harsh/Abrupt Endings" criteria show particularly impressive results with 90% and 91% accuracy respectively, suggesting these prosodic features are more distinctly encoded in the sequential acoustic patterns. The "Service-Minded Tone" achieves 84% accuracy with moderate F1 (0.77) and AUC (0.82) scores, while "Non-Sluggish Speech" shows the lowest accuracy at 76% but maintains competitive F1 (0.84) and AUC (0.88) metrics. This pattern indicates that while individual models can effectively learn the nuances of each criterion independently, they may not fully leverage the relationships between different tone qualities, resulting in a solid baseline performance averaging 86.5% across criteria.

The Multitask model approach shows consistent improvement over individual models, with accuracy increases across all criteria and an overall average of 86.5%. By sharing representations between related tone criteria through a common feature extraction backbone, the multitask architecture effectively transfers knowledge across tasks, leading to improved generalization. The "Service-Minded Tone" and "Non-Sluggish Speech" criteria benefit most from this approach, with accuracy improvements of 2 percentage points each (to 86% and 78% respectively), suggesting these criteria share underlying acoustic patterns that can be mutually reinforced. The "Dynamic Intonation" criterion shows a smaller improvement to 91% accuracy, while "No Harsh/Abrupt Endings" maintains its 91% performance. The multitask model achieves comparable or slightly better F1 scores across criteria, indicating that the shared representation learning does not compromise the balance between precision and recall. This performance validates the hypothesis that tone criteria are interrelated aspects of speech prosody rather than entirely independent qualities.

The Multitask with Separate SMOTE approach delivers the strongest overall performance, achieving an average accuracy of 87.8% and setting the highest scores for each individual criterion.

This enhanced architecture addresses class imbalance through specialized SMOTE techniques adapted for the multitask context, creating synthetic samples that balance all criteria simultaneously while preserving their natural correlations. The most notable improvement appears in "Dynamic Intonation," which reaches 93% accuracy with exceptional F1 (0.95) and AUC (0.97) scores. "Service-Minded Tone" also benefits substantially, achieving 87% accuracy with improved F1 (0.80) and AUC (0.85) metrics. "Non-Sluggish Speech" maintains its 78% accuracy but shows slightly improved F1 (0.85) and AUC (0.89), while "No Harsh/Abrupt Endings" holds steady at 91% accuracy with consistent F1 and marginally higher AUC (0.94). These results demonstrate that addressing class imbalance in a way that respects the multitask structure provides additional performance benefits beyond just sharing representations, resulting in the most accurate and balanced tone analysis system.

4.6 Criteria 1.7

For Criteria 1.7, we employ an appropriate word usage system for bad word detection and negative content detection, utilizing WangchanBERTa, a model with 91% accuracy, combined with POS tagging for enhanced performance. To evaluate the system, we evaluate using our synthesized data, which may introduce some bias. Despite this limitation, the model achieved an accuracy of 99% in our evaluation.

4.7 Criteria 1.9

The result is divided into two parts: the transliterated word result and the specialized word result. For the transliterated word result, it is rule-based, so it will always be correct if the text is transcribed accurately. For the specialized word result, *paraphrase-multilingual-MiniLM-L12-v2*, which is an open source model, is used to determine the similarity level. This model has an average Spearman correlation of around 0.8 on the STS Benchmark and Multilingual STS.

4.8 Criteria 1.11

The result of this criterion depends on two main factors: model accuracy and the ideal ending phrase. For model accuracy, *paraphrase-multilingual-MiniLM-L12-v2*, which is an open source model, is used to determine the similarity level. This model has an average Spearman correlation of around 0.8 on the STS Benchmark and Multilingual STS. As for the ideal ending phrase, it must be set by the stakeholder, and it can directly affect the similarity.

Chapter 5
Conclusions

5.1 Summary of Accomplishments

Our accomplishments include data synthesis, speaker diarization, development and fine-tuning of a speech-to-text model, creation of a model pipeline, and implementation of the targeted evaluation criteria. Upon completing the model pipeline, we finalized the specific criteria to be addressed. Achieving this required extensive research in both text and tone analysis to determine the feasibility and scope of coverage, ensuring the pipeline's overall viability.

Additionally, the speech-to-text model was fine-tuned and tested, resulting in improved performance with a lower word error rate. This component is critical, as most evaluation criteria rely on accurately transcribed text. Any errors in transcription could lead to incorrect assessments, making the quality of the speech-to-text model essential to the overall system’s effectiveness.

5.2 Issues and Obstacles

Most of the issues we faced this semester were beyond our control and occurred due to the lack of available datasets, primarily because of privacy concerns. This resulted in bias in the synthesized data, as no actual labeled dataset was provided.

5.2.1 Data Access

Although the Memorandum of Understanding (MOU) has been finalized, the dataset has not yet been provided due to ongoing privacy concerns. The data may contain sensitive personal information and require legal verification. Without access to a labeled dataset, it is impossible to determine whether it meets the necessary criteria. As a result, training the model has been extremely challenging.

5.2.2 Bias in Synthesized Data

To address the absence of a real dataset, we synthesized our own using the Qwen2.5-72B model. The NHSO provided 11 sample summaries (not actual data), which were used—along with the given criteria—to generate a new dataset from scratch. However, due to the limited number of samples, the synthesized data tends to share a similar context with only minor variations, potentially introducing bias. There is also bias in our tone dataset and label. As the label heavily relied on emotion mapping and crowdsourcing, it isn’t made specifically for tone. Emotion includes every aspect of the recording, tone, speech rate, words usage, the intonation depending on each word’s context, and more.

5.2.3 Model limitations

Although the STT model has been fine-tuned, it still exhibits a high word error rate (WER). The STT model plays a crucial role; therefore, a 16% WER is considered significant. One of the most important reasons for this is that the model is applied to the Thai language, which presents

unique linguistic challenges such as tonal variations, lack of word boundaries, and limited training data compared to more widely spoken languages.

On the other hand, the speaker diarization model performs well when the speakers are of different genders. However, if the speakers share the same gender, the diarization error rate increases significantly. This could potentially lead to misidentification between the agent and the citizen, resulting in the text and tone analysis being attributed to the wrong individual.

5.3 Future Directions

For the program to effectively replace human assessment, it must first be capable of covering a broader range of evaluation criteria. Therefore, a key future direction is to expand its scope to accommodate additional criteria. Furthermore, fine-tuning the model with real-world datasets will be essential, as synthetic data cannot fully capture the variability and noise present in actual data.

5.3.1 Expand the Criteria Scope

According to the NHSO, the evaluation framework consists of four main sections, each containing multiple sub-criteria. Currently, our program covers only half of the first section, which is insufficient to replace or even meaningfully assist the supervisor and QA team. Therefore, the scope of the criteria must be significantly expanded for the system to be applicable in real-world scenarios.

5.3.2 Implementation with Real Dataset

The current models have been trained, evaluated, and tested using synthesized data, which may not fully capture the noise and variability present in real-world scenarios. Therefore, once the actual dataset becomes available, we plan to retrain and fine-tune the models to enhance their ability to perform accurately under unpredictable and noisy conditions. We would also have to reanalyze the data and the label from our stakeholder to validate our tone analysis approach. If it is, then we will focus on optimizing the threshold of each criteria to best suit the stakeholder's standard.

5.3.3 System integration

At this stage, we have developed a standalone website; however, the program will ultimately need to be deployed within the NHSO's infrastructure. There are no concerns regarding computing power, as the program can run efficiently on a CPU. The current run time corresponds with the audio's real duration. Optimizing the pipeline and running the system on GPU would most likely decrease the run time. Once development is complete, the next step will be to integrate the system into the NHSO's existing infrastructure.

5.3.4 Additional feature in Criteria 1.5 - tone and text analysis

If tone analysis from recording alone proves to be insufficient, we will explore combining tone and text analysis even more in the future. We won't only search for twang or ending particles,

but possible sentiment and context analysis as well to increase the model's understanding and overview of the situation.

5.4 Lessons Learned

Our lessons learned are divided into two main aspects: soft skills, which involve working with external organizations, and hard skills, which focus on the technical aspects.

5.4.1 Cooperation with External Organization

Working with external organizations has been an important lesson, as not everything is within our control, such as access to datasets. It is essential to understand their perspective and maintain regular communication to ensure the work progresses as efficiently as possible.

5.4.2 Speech-To-Text, Text Analysis, and Tone Analysis

Another lesson we have learned is related to technical aspects, particularly in Speech-to-Text, Text Analysis, and Tone Analysis. Developing a reliable Speech-to-Text system involves addressing challenges like varying accents, background noise, and speech clarity to improve accuracy.

We have conducted extensive research on text analysis, as it is used to evaluate the majority of our criteria. As a result, we have developed a strong understanding of how both rule-based and model-based approaches work. Furthermore, we have recognized that it is not always necessary to rely on large language models (LLMs), as they require significant computing resources. There are alternative solutions that offer acceptable accuracy while maintaining low computational demands.

As for tone analysis, none of us had prior experience in this topic, resulting in an intensive research and experimentation. We have learned both soft skills - improving our problem solving skills every time an obstacle arises as well as time management - and technical skills where we dived down to the fundamentals up. We went through basic classifiers, different techniques in labeling a dataset, advanced models like LSTM, and differences in individual and multitask models. Working from bottom up gave us the opportunity to observe the difference between the classic and more simplistic models compared to the more advanced ones, and solidified our reasoning behind our final approach choices. We've learned to be thorough in each step, planning and implementing, after having to go back and redo multiple steps in the end. Most importantly, we learned when to call it quit. Working with both time and technical constraints, we learned when to decrease the scope's size and re-evaluate what we can do. We learned a great deal about perseverance in this project.

References

[1] *Philosophy & Background*. Philosophy Background. (n.d.). https://eng.nhso.go.th/view/1/Philosophy_Background/EN-US

[2] Hu, C., & Downie, A. (2024, November 25). *What is speech to text?*. IBM. https://www.ibm.com/think/topics/speech-to-text

[3] Manakul, P. (n.d.). *Audio Preview release*. Typhoon. https://opentyphoon.ai/blog/en/typhoon-audio-preview-release-6fbb3f938287

[4] Rkcosmos. (n.d.). *RKCOSMOS/Deepcut: A Thai word tokenization library using Deep Neural Network*. GitHub. https://github.com/rkcosmos/deepcut

[5] Jitsi. (n.d.). *Jitsi/jiwer: Evaluate your speech-to-text system with similarity measures such as word error rate (WER)*. GitHub. https://github.com/jitsi/jiwer

[6] R&D, L. J. (2023, July 17). *Speaker diarization: An introductory overview*. Medium. https://lajavaness.medium.com/speaker-diarization-an-introductory-overview-c070a3bfea70

[7] Pyannote. (n.d.). *Pyannote/pyannote-audio: Neural building blocks for speaker diarization: Speech activity detection, speaker change detection, overlapped speech detection, speaker embedding*. GitHub. https://github.com/pyannote/pyannote-audio

[8] YouTube. (n.d.). YouTube. https://www.youtube.com/watch?v=37R_R82lfwA

[9] *Regex tutorial: Learn with regular expression examples*. 4Geeks. (2024, January 10). https://4geeks.com/lesson/regex-tutorial-regular-expression-examples

[10] *RE - regular expression operations*. Python documentation. (n.d.). https://docs.python.org/3/library/re.html

[11] Auto QA for call centers | automation in call center quality assurance. (n.d.). https://www.scorebuddyqa.com/blog/auto-qa-call-centers-automation

[12] Shrivastav, R. R. (2024, July 24). *Understanding call recording processes in modern voice call centers*. Understanding Call Recording Processes in Modern Voice Call Centers. https://convin.ai/blog/voice-calls-recorder

[13] Forbes Magazine. (2024, October 15). *Call Center Analytics Guide (2024)*. Forbes. https://www.forbes.com/advisor/business/call-center-analytics/?utm_source=chatgpt.com

[14] Hugging Face. (n.d.) https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

[15] AIModels.fyi. (n.d.) | all-MiniLM-L12-v2

https://www.aimodels.fyi/models/huggingFace/all-minilm-l12-v2-sentence-transformers

[16] GeeksforGeeks. (2025a, April 8). *Natural language processing (NLP) - overview*. https://www.geeksforgeeks.org/natural-language-processing-overview/

[17] GeeksforGeeks. (2024, January 3). *Pos(parts-of-speech) tagging in NLP*. https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/

[18] VISTEC-depa AI Research Institute of Thailand. (2021, February 24). *Wangchanberta โมเดลประมวลผลภาษาไทยที่ใหญ่และก้าวหน้าที่สุดในขณะนี้*. Medium. https://medium.com/airesearch-in-th/wangchanberta-/

[19] *Pythainlp*. PyPI. (n.d.). https://pypi.org/project/pythainlp/

[20] Recent developments in openSMILE, the Munich open-source multimedia feature extractor | Proceedings of the 21st ACM international conference on multimedia. (n.d.). https://dl.acm.org/doi/10.1145/2502081.2502224

[21] *Introduction to audio analysis*. ScienceDirect. (n.d.). https://www.sciencedirect.com/book/9780080993881/introduction-to-audio-analysis

[22] Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences | IEEE Journals & Magazine | IEEE Xplore. (n.d.). https://ieeexplore.ieee.org/document/1163420/

[23] IRCAM. (n.d.-b). http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

[24] *Archiveredirection*. ISCA. (n.d.). https://www.isca-speech.org/archive/interspeech_2011/i11_1973.html

[25] UVA. (n.d.-c). https://www.fon.hum.uva.nl/paul/papers/Proceedings_1993.pdf

[26] Meltzer, R., Rachel Meltzer Writer for The CareerFoundry BlogRachel is the founder of MeltzerSeltzer, Rachel Meltzer Writer for The CareerFoundry Blog, Meltzer, R., Blog, W. for T. C., & Rachel is the founder of MeltzerSeltzer. (2023, August 31). *What is Random Forest? [beginner's guide + examples]*. CareerFoundry. https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/

[27] What is RNN? - recurrent neural networks explained - AWS. (n.d.-d). https://aws.amazon.com/what-is/recurrent-neural-network/

[28] *Understanding LSTM networks*. Understanding LSTM Networks -- colah's blog. (n.d.). https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[29] *Smote: Synthetic data generation for balanced datasets*. Lyzr. (2025, January 8). https://www.lyzr.ai/glossaries/smote/

[30] Caruana, R. (2016, February 19). *Multitask learning - machine learning*. SpringerLink. https://link.springer.com/article/10.1023/A:1007379606734

[31] Ruder, S. (2017, June 15). *An overview of multi-task learning in Deep Neural Networks*. arXiv.org. https://arxiv.org/abs/1706.05098