Master Thesis

Exploring An Existing ASR Model for a Binary Classification of Intelligibility on MOOC English Speech Data

Furong Zou

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by:	Luis Morgado da Costa, Pia Sommerauer
2nd reader:	L.E. Donatelli
Submitted:	August 20, 2024

Abstract

This thesis investigates the feasibility of using Whisper's English-only base model (*base.en*) to detect pronunciation errors in spoken English data from the Massive Open Online Course (MOOC) *English Pronunciation in a Global World* that is provided at VU Amsterdam. Given the challenges of providing individualized pronunciation feedback in large-scale online courses, there is a need for more reliable and automated assessment tools. This study focuses on developing a classifier for word-level binary classification of intelligibility by leveraging speech-to-text transcriptions generated by Whisper. Various temperature settings and confidence thresholds are explored to optimize the model's accuracy in identifying mispronounced words. A phonetic-based classification method is also employed to handle homophones effectively, ensuring that the model can distinguish between intelligible and unintelligible pronunciations despite of the spelling. The performance of the classifier is evaluated against gold standard annotations provided by a linguistic expert.

Experiments revealed that while temperature settings had a minimal effect on the model's precision, higher temperatures increased the frequency of nonsensical transcriptions. Confidence thresholds demonstrated a clear trade-off between precision and recall, where stricter thresholds reduced the model's ability to detect unintelligible speech. The phonetic-based classification method consistently outperformed the text-based approach, suggesting it may offer a more robust solution for intelligibility tasks.

Despite these insights, the Whisper model's performance in word-level intelligibility classification fell short of expectations. The model's tendency to produce near-homophones for errors rather than random inaccuracies highlights both its strengths and limitations. Overall, the findings indicate that while Whisper shows promise in speech recognition, its current design is not fully suited for accurate pronunciation error detection. Further research into alternative ASR models and refined methodologies is recommended to enhance automated pronunciation assessment systems.

Declaration of Authorship

I, Furong Zou, declare that this thesis, titled *Exploring An Existing ASR Model for a Binary Classification of Intelligibility on MOOC English Speech Data* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master's degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: August 20, 2024

Signed: Furong Zou

Acknowledgments

I would like to express my deepest gratitude to my supervisors, Luis Morgado da Costa and Pia Sommerauer, for their invaluable guidance, continuous support, and insightful feedback throughout this thesis project. Their expertise and encouragement were instrumental in navigating the challenges of this study and pushing me to complete my work.

I would also like to thank my colleagues and peers, whose helpful discussions, shared resources, and moral support during this thesis project and throughout the entire Master's program have greatly enriched my experience.

Special thanks to my husband, Qinghao Shen, and my family for their love, unconditional support, and understanding during the many hours I spent working on this project. Without them, I would not have been able to achieve this.

Finally, I am grateful to the CLTL staff at VU Amsterdam for sharing their knowledge and providing such a collaborative environment for academic growth.

List of Figures

Contents

Chapter 1

Introduction

1.1 Problem Definition

English Pronunciation in a Global World[1] is an online course provided by the *Centre for Global English* at VU Amsterdam. Like other Massive Open Online Courses (MOOC), it aims to help a diverse global audience improve their English language skills. One of the key components of this course is spoken English, where learners submit speech samples for assessment. However, the sheer volume of participants in such an online course makes it challenging to provide individualized feedback on pronunciation. Traditional methods of pronunciation assessment rely heavily on human evaluators, which are time-consuming, leading to challenges in providing personalized feedback, especially in large-scale learning environments where hundreds or even thousands of learners participate simultaneously.

The course *English Pronunciation in a Global World* now mainly relies on peer feedback that is made by non-native speakers of English, which can be subjective and inconsistent, leading to less precise feedback on pronunciation. However, accurately assessing pronunciation is crucial for supporting learners' progress in language learning. This is also the reason why *English Pronunciation in a Global World* is now seeking solutions to supplement peer feedback with more reliable tools or methodologies that automate pronunciation assessment. While peer feedback fosters a collaborative learning environment, it may not always provide the level of accuracy needed for learners to make significant progress. Given this consideration, there is a pressing need for a more efficient and objective method (e.g., automated corrective feedback system) to assess pronunciation in such contexts. The course could benefit from integrating such systems that are developed by using expert annotation of speech data to ensure that learners receive precise and constructive guidance on their pronunciation skills.

This internship-based thesis aims to explore and evaluate the effectiveness of detecting pronunciation errors in an existing automatic speech recognition (ASR) model, specifically Whisper's English-only base model (*base.en*). The focus will be on performing a speech-to-text transcription of English speech data from the MOOC and using these transcriptions to build a classifier used for a word-level binary classification of intelligibility. The ultimate goal is to determine whether Whisper can be effectively used as a foundation for an automated corrective feedback system (which will not be implemented in this project). The classification of intelligibility is the first essential step towards providing such automated feedback, making this research highly relevant for improving the quality of language learning in MOOC.

[1]https://www.futurelearn.com/courses/english-pronunciation

1.2 Research Question

This project explores the feasibility of using Whisper's English-only base model (*base.en*) to detect pronunciation problems in MOOC English speech data. Specifically, the primary research question is:

Can Whisper be used for detecting pronunciation problems by checking the intelligibility of speech through a word-level binary classification?

1.3 Approach

The project focuses on comparing the performance of the same model (*base.en*) with different temperature settings and confidence thresholds to find specific parameter values that optimize the model's ability to effectively identify mispronounced English words.

This is how it works: Firstly, the Whisper model is used to get transcriptions for word-level speech data. Secondly, by comparing the model-generated transcriptions with the true transcriptions (the written word), a classifier is built for assigning labels, for instance, if the model's transcription is identical to the target word, a label of 'intelligible' will be assigned to this input speech data, otherwise 'unintelligible'. To better handle homophones, a phonetic-based classification method using CMUDict is also applied. Taking the word 'meet' as an example, if the model generates 'meat' as the transcription, it will still receive an 'intelligible' label based on the phonetic-based classification method. Finally, these labels are compared with the gold labels (gold annotation of intelligibility by a linguistic expert) to evaluate the model's performance of classifying intelligible and unintelligible instances.

Different temperature settings are set when using the model to do speech-to-text conversion, and the confidence scores are obtained along with the transcription and then being used as a threshold. By exploring various configurations of temperature settings, confidence scores, and classification methods, the optimal model will be found and then run on the test data.

1.4 Outline of the Thesis

The chapters of this thesis are segmented as follows. Chapter 2 provides an overview of previous related work, focusing on automatic pronunciation checkers. Chapter 3 elaborates the methodology (including the data and experimental setup) used for this task, followed by Chapter 4 (Results) and Chapter 5 (Error Analysis). Chapter 6 provides a discussion and insights into this project's limitations and future directions. Lastly, Chapter 7 concludes this thesis project.

Chapter 2

Related Work

This chapter provides an overview of recent related work in educational Natural Language Processing (NLP) with a specific focus on applications in language learning and automatic feedback systems, particularly for pronunciation. It surveys existing literature and technologies, highlighting key approaches such as automatic pronunciation checkers integrated with Automatic Speech Recognition (ASR).

2.1 Educational NLP for Language Learning

Natural Language Processing (NLP) has many advantages in education as it enables personalized language learning experiences (Younis et al., 2023). It plays a crucial role in language learning by enhancing the way learners interact with and understand new languages. Through technologies like automated translation (e.g., Google Translate) which facilitates communication and understanding across languages, spelling or grammar correction (e.g., Grammarly), and conversational chatbots or virtual assistants (e.g. Siri, Alexa, Google Assistant), NLP allows learners to practice language skills in real-time, receive instant feedback, and access personalized learning content. By analyzing and generating human language, NLP helps learners learn complex linguistic patterns, improve their fluency, and engage more deeply with the language in a dynamic, interactive manner.

For now, NLP is widely integrated with a large number of educational contexts such as research, science, linguistics, e-learning, and evaluation systems, and contributes to resulting positive outcomes in other educational settings such as schools, higher education systems, and universities (Alhawiti, 2014; Mathew et al., 2021).

Pronunciation, as a critical aspect of language learning that directly impacts a learner's ability to communicate effectively in the target language, has been increasingly supported by advanced technologies and methodologies. Popular language learning platforms like Duolingo[1], Babbel[2], and Rosetta Stone[3] are developed using NLP to personalize learning, assess language proficiency, and even provide real-time feedback.

[1]https://en.duolingo.com/
[2]https://uk.babbel.com/
[3]https://eu.rosettastone.com/

2.2 Automated Pronunciation Feedback Approaches

This section reviews some related works of automatic pronunciation checkers for language learning.

For detecting pronunciation errors, various techniques have been used in computer-aided pronunciation training (CAPT) in English, including visual simulation-based systems that record and analyze the speech of a learner, and then provide feedback through visual aids (e.g., explanatory images, or comparative videos) (Kalikow and Swets, 1972), comparative phonetics based systems where the phonemes in the native language of a learner are compared to those in English using a probabilistic analysis (Wang et al., 2008; Qian et al., 2016), and recently deep neural networks based systems (Li et al., 2016; Wang and Lee, 2015; Upadhyay et al., 2021).

Besides these techniques mentioned, ASR has also been employed in computer-assisted language learning (CALL) in the field of pronunciation learning, with two primary objectives: to teach the correct pronunciation of a foreign language to learners, and to assess the pronunciation quality of individuals speaking that language (pronunciation scoring). Here below are related previous works utilizing ASR for various language learning, mainly focusing on the first goal.

Jo et al. (1998) proposed a learning system to detect pronunciation errors and provide diagnostic feedback through speech processing and recognition methods to aid non-native speakers studying Japanese language as a second language. Kawai and Hirose (2000) also developed a CALL system for teaching the pronunciation of Japanese double-mora phonemes to non-native speakers of Japanese, in which speech recognition algorithms were applied to accurately measure the duration of *tokushuhaku*, ‘a set of phonemically distinct phones most non-native learners have difficulty with’.

Abdou et al. (2006) developed a system for teaching Arabic pronunciations to non-native speakers, by using a state-of-the-art speech recognizer to detect errors in the user’s recitation.

(Strik et al., 2009) compared four types of classifiers that can be used to detect erroneous pronunciations of the velar fricative /x/ and the velar plosive /k/ in Dutch, which are pronunciation errors frequently made by foreigners learning Dutch as a second language.

With the object of teaching correct American English pronunciation (specifically pronunciation of basic English phonemes) to high-school students in Hong Kong, whose native language is Cantonese, Mak et al. (2003) designed a multimedia learning tool called PLASER (Pronunciation Learning via Automatic SpEech Recognition) with instant feedback.

ASR works by analyzing speech input from a speaker, and comparing it, usually with a native speaker model, which is created using a database filled with numerous samples of native speaker recordings (Rogerson-Revell, 2021). Rogerson-Revell (2021) also mentioned the significant potential of ASR technologies to provide immediate, personalized feedback.

Chapter 3

Methodology

This chapter provides a detailed description of the methodology employed in the project. Three main sections are included: Task Explanation, Data, and Experimental Setup.

Section 3.1 explains the task and its goals. Section 3.2 (Data) describes the MOOC speech data and details of audio data preprocessing, including the steps taken to clean, segment, and prepare the audio data for input into the Whisper model. Section 3.3 (Experimental Setup) covers the rationale behind choosing Whisper's English-only base model for the task, the temperature parameter tuning, the collection and post-processing of Whisper-generated transcription data, the confidence threshold, the classification algorithms and evaluation metrics used.

3.1 Task Explanation

The primary objective of this project is to evaluate the efficacy of OpenAI's Whisper English-only model (*base.en*) as a tool for detecting mispronounced English words. Specifically, the task involves a word-level binary classification of intelligibility, where the goal is to determine whether the Whisper model can accurately identify whether a spoken English word is pronounced intelligibly by non-native speakers.

3.2 Data

3.2.1 MOOC Data

Raw audio recordings. The primary dataset for this project consists of audio recordings obtained from students enrolled in *MOOC English Pronunciation in a Global World*[1]. These recordings are provided in MP3 format and each contains a student reading a list of 52 English words (for some of the recordings, a reading passage is included as the assignment for this course consists of reading the Word List and the Reading Passage). They were accessed using URLs listed in Excel files facilitated by Laura Rupp, the director of the *Centre for Global English* at VU Amsterdam and the creator and instructor of the online course. Importantly, all recordings were obtained with the informed consent of the students, ensuring ethical standards were maintained in data usage.

Expert-labelled test set. A subset of 45 raw audio recordings recorded by 45 students, referred to as *participant-level recordings*, has been selected and meticulously annotated at the word level by linguistic expert Laura Rupp to serve as the test set for the project. These

[1]https://www.futurelearn.com/courses/english-pronunciation

participant-level recordings are balanced in terms of recording quality, ranging from unintelligible to highly intelligible. Based on the overall intelligibility of each recording, the test set initially included 15 highly intelligible, 15 fairly intelligible, and 15 poorly intelligible *participant-level recordings*. Although the experiments were ultimately conducted at the word level (preprocessed MP3 files, referred to as *word-level audio files*) rather than the student level (*participant-level recordings*), this information is retained to clarify the selection of 4 *participant-level recordings* from the test set as development data (further explained in Development set).

In the Excel file used for labeling the test set, the intelligibility of individual words is categorized into four classes: 'yes', 'no', 'different English vowel', and 'mispronounced' by the expert. For this binary classification task, the classes 'different English vowel' and 'mispronounced' are combined with 'no' to form a single 'unintelligible' class, while 'yes' remains as 'intelligible'. The class 'different English vowel' is actually a particular type of error that causes an 'unintelligible' impression of the spoken word, being treated differently by the expert during the annotation. This may be due to the common difficulty of second language learners pronouncing vowel sounds, especially influenced by his or her native language (Bada, 2001), for instance, Japanese speakers may tend to shorten English long vowels which do not exist in their native language's phonological system. In addition, Flege et al. (1997) also found that the experienced non-native subjects who were exposed intensively to English produced and perceived English vowels more accurately than the relatively inexperienced non-native subjects.

In some *participant-level recordings*, some words are either not pronounced or repeated by the speaker. These words are marked as 'not pronounced' or 'correction' in the Excel file and are excluded from the test set as they lack the gold standard label. Consequently, the final test set consists of 2,115 word-level MP3 files, which were derived from 41 participant-level recordings.

Development set. I selected 6 relatively clean recordings with minimal background noise from the raw audio data for annotation. In addition, I included 4 *participant-level recordings* from the test set, comprising one highly intelligible, one fairly intelligible, and two poorly intelligible recordings. The reason for this selection is to ensure that the development set includes a enough proportion of unintelligible words, which is essential for evaluating and improving the model's robustness and accuracy in handling varied intelligibility levels. These 10 *participant-level recordings* constitute the development dataset. After preprocessing, they correspond to 516 *word-level audio files* and will be used to build the pipeline and tune the model's parameters to optimize performance across different speech clarity levels.

Text data: true transcriptions. The 52 written words in Word List (shown below in Table 3.1) are given and served as the true transcriptions.

3.2.2 Audio Data Preprocessing

Audio data preprocessing is the most time-consuming yet crucial step for achieving precise analysis in this task. The most important step of audio data preprocessing is segmenting *participant-level recordings* into *word-level audio files* since there is a known limitation of Whisper's sequence-to-sequence architecture, which is prone to generating repetitive texts, as noted on Hugging Face [2]. For example, when dealing with a *participant-level recording* containing 52 words — the same number as in the Word List, the model tends to generate more written words than were actually spoken by the speaker, complicating the task of aligning the

[2]https://huggingface.co/openai/whisper-base.en

1. pit	14. beer	27. pull	40. plate
2. pet	15. bear	28. pool	41. weight
3. pat	16. bird	29. pole	42. poor
4. put	17. bard	30. paul	43. pour
5. putt	18. board	31. doll	44. pore
6. pot	19. city	32. cot	45. paw
7. bee	20. seedy	33. caught	46. tide
8. bay	21. hat	34. fir	47. tied
9. buy	22. dance	35. fern	48. pause
10. boy	23. daft	36. fur	49. paws
11. boot	24. half	37. fair	50. meet
12. boat	25. father	38. nose	51. meat
13. bout	26. farther	39. knows	52. mate

Table 3.1: Word List

transcriptions. This situation occurred frequently when I used the model to do speech-to-text transcription on the development data (*participant-level recordings*).

I initially attempted to automate the segmentation process by using Whisper's function[3] *whisper_timestamped.transcribe()* to obtain the start and end times for each word, with the goal of automatically segmenting the recordings. However, the timestamps provided by the model were 'phrase-level timestamps'[4], as noted by Whisper, which were too coarse for accurate word-level segmentation. To address this, then I tried using *whisper-timestamped*[5], an extension of the OpenAI Whisper Python package that offers relatively more accurate word-level timestamps. Despite this improvement, the timestamps obtained by *whisper-timestamped* were still not precise enough for reliable segmentation. This approach proved insufficient after I listened to the segmented audio files because it failed to get complete English words. For example, the word 'pit' was truncated to 'pi', with the final consonant missing, indicating that the word boundaries were not correctly identified. It also struggled to accurately distinguish between the target words and additional, unintended speech segments.

Given these limitations, manual segmentation became necessary to ensure accuracy. This manual process involves removing the non-relevant passage reading part and precisely segmenting the Word List recordings into word-level MP3 files. Since this project only utilizes the speech segments from the Word List, any non-relevant passage reading parts (mentioned in Subsection 3.2.1 Raw audio recording) have been removed.

Besides the known limitation mentioned, segmenting recordings into individual words is motivated by other three primary considerations. Firstly, it helps reduce noise in the audio data, as students may give an introductory speech before reading the list of words or produce repeated corrective utterances for the same word. Secondly, a significant challenge in this task involves handling near-homophones and homophones. The Word List mainly consists of these English word types, both of which can lead to confusion in speech or writing, also during transcription. Near-homophones are pairs or groups of words that sound very similar but differ slightly in pronunciation (e.g., 'city' and 'seedy', where only one sound differs), with different spellings and meanings. Unlike homophones, which are pronounced the same way, near-

[3]https://github.com/linto-ai/whisper-timestamped?tab=readme-ov-file#python
[4]https://openai.com/index/whisper/
[5]https://pypi.org/project/whisper-timestamped/

homophones require careful attention to detail because the subtle differences in pronunciation are critical for conveying the correct meaning. These distinctions are particularly important in pronunciation practice, as students must learn to accurately pronounce and differentiate these words to avoid misunderstandings. In this task, it is essential that near-homophones are transcribed correctly, ensuring that the model can distinguish between them based on these minor but significant variations in sound.

Homophones pose a special and inherent challenge in this task and other pronunciation tasks that involve converting spoken language into text because they are words that sound the same but have different spellings and meanings (e.g., 'meet' and 'meat'), which can create confusion. This problem is different from other challenges from the quality or clarity of the speech data, such as background noise or poor recording quality, which are external factors that can be addressed to some extent through preprocessing. Instead, the homophone issue is an intrinsic difficulty in any task where spoken words need to be accurately transcribed into text. For instance, when the model encounters a word that sounds the same as another word (a homophone), it has to decide which word to transcribe based on the context. However, for this task where the data is individual words, even if the model perfectly captures the pronunciation, it may choose the wrong word since there is no context information.

In addition, pairs or groups of near-homophones and homophones appear consecutively in the Word List, making it more challenging to align the transcriptions. Although the homophone issue is a problem specific to the nature of text-related pronunciation tasks rather than an irregularity in the data, it still complicates alignment. Thus, segmenting the recordings into word-level audio files facilitates a more accurate alignment between Whisper's transcriptions and the true transcriptions.

Thirdly, for getting the word-level confidence score as a threshold (will be explained in Subsection 3.3.4) to test the model, precisely splitting audio files into individual words is needed because the Whisper model only provides segment-level confidence rather than word-level confidence. For example, for a *participant-level recording*, the model generally transcribed it into several segments due to its internal processing logic: reading the entire audio file and then processing the audio with a sliding 30-second window to make autoregressive sequence-to-sequence predictions[6]. After transcribing, each segment contains the text and the corresponding confidence. To be specific, below is an example of the text that belongs to the first segment of a transcription given for a *participant-level recording*. The model provides the confidence information for the entire text rather than for individual words, which means all the words share the same value of confidence, leading to an inaccuracy measure of how confident the model is in its transcriptions for individual words.

- *' Pit, pit, pat, put, put, put, put, put, be, be, by, boy, boot, boat, boat, beer, bear,'*

Considering the above reasons, manually preprocessing the audio files became necessary. The distribution of the classes in the preprocessed development audio dataset and test audio dataset is shown in Table 3.2, showing a predominance of the 'intelligible' class over the 'unintelligible' class in both datasets.

Audacity, an open-source software for recording and editing audio, was used for audio data preprocessing. Given the need to name the audio files in a way that helps trace back to the words and their gold labels, a naming convention that encapsulates the relevant information was created for better traceability, structured as **studentID_word_trueLabel**. To be specific, *studentID* is the unique identifier for each student (e.g., s01, s02, ..., s51), *word* is the actual

[6]https://github.com/openai/whisper

Dataset	Intelligible	Unintelligible	Total Instances
Development	353 (68.4%)	163 (31.6%)	516
Test	1410 (66.7%)	705 (33.3%)	2115

Table 3.2: Distribution of Classes Intelligible and Unintelligible in Development and Test data

word being pronounced (referred to as the true transcription), and *trueLabel* is a binary numerical representation added based on gold annotation. Specifically, 1 is for denoting intelligible and 0 for unintelligible. After segmenting, all *word-level audio files* were named following this convention.

3.3 Experimental Setup

This section outlines the overall intuition behind the use of a speech-to-text model for this task, followed by the details of the experimental design, including model selection, model parameter tuning, collection and post-processing of transcriptions and relevant information, confidence threshold setting, the classification algorithms and the evaluation metrics used.

3.3.1 Overall Intuition and Approach

The core idea behind this project is to leverage the ASR system, specifically Whisper's English-only base model to transcribe audio recordings of English words for a word-level binary classification of intelligibility, trying to find if the model is a suitable tool used in assessing pronunciation in language learning. This approach is motivated by the need for efficient, precise, and scalable analysis of English speech data, which is critical for assessing and improving the intelligibility of the spoken language. Traditional methods of pronunciation assessment rely heavily on human evaluators, which are labor-intensive, leading to challenges in providing personalized feedback, especially in large-scale learning environments.However, ASR models provide the possibility to automate the pronunciation assessment process, offering consistent and objective feedback to language learners.

The approach involves using Whisper's ASR capabilities to transcribe spoken words into text and then perform a word-level binary classification of intelligibility. By comparing the transcriptions generated by Whisper with the true transcription, we establish an intelligibility classifier that reflects the model's assessment of intelligibility. This classifier serves to evaluate the model's capability in identifying mispronounced words through its binary classification.

3.3.2 Model Selection

The Whisper model is well-suited for this task due to its robustness and strong zero-shot transfer capabilities. Whisper's models, trained on a broad and diverse distribution of audio data with comprehensive 680,000 hours of multilingual and multitask supervision, demonstrate remarkable generalization abilities to standard benchmarks, often rivaling the results of models trained in a fully supervised manner, all without any fine-tuning and purely in a zero-shot transfer setting (Radford et al., 2023). The strong zero-shot transfer capabilities can make the model perform well on new datasets without the need for fine-tuning on specific datasets, which is beneficial when assessing pronunciation across diverse accents and proficiency levels. In addition, Whisper models are robust to background noise, especially high noise. Unlike many other models that quickly degrade in accuracy under high noise conditions, Whisper maintains high transcription accuracy even in noisy settings. This resilience ensures that the model can provide reliable results regardless of the recording conditions or background noise, a crucial aspect for language learning applications where audio quality can vary significantly.

According to the documentation, Whisper offers five model sizes, with four of them available in English-only versions, providing a trade-off between speed and accuracy. The names of the available models and their specifications are listed in Table 3.3.

Size	Parameters	English-only model	Multilingual model	Relative speed
tiny	39 M	*tiny.en*	tiny	∼32x
base	74 M	*base.en*	base	∼16x
small	244 M	*small.en*	small	∼6x
medium	769 M	*medium.en*	medium	∼2x
large	1550 M	N/A	large	1x

Table 3.3: Comparison of Whisper model sizes and specifications

The choice of the *base.en* model for speech-to-text transcription in this project is influenced by insights from Whisper's documentation[7]. Initially, it is observed that models suffixed with *.en*, tailored for English-only applications, tend to perform better, especially for the *tiny.en* and *base.en*. This advantage may stem from the fewer parameters which make the models perform better in monolingual situations. I opted for the English-only models since the audio data is in English, aiming to achieve higher accuracy. Additionally, observations from the OpenAI Whisper team indicate that the difference becomes less significant for the *small.en* and *medium.en* models.

3.3.3 Model Parameter Tuning: Temperature

Temperature is an optional parameter in the ASR model that ranges from 0 to 1. This parameter controls the degree of variability or 'creativity' of the model's output: a higher temperature value (closer to 1) allows the model to generate a broader range of outputs, reflecting more variability in pronunciation and word choice. Conversely, a lower value (closer to 0) makes the model's output more focused and deterministic.

In this intelligibility prediction task for audio data made by non-native English speakers, adjusting the temperature parameter is crucial for managing the balance between capturing the nuances of natural variability in speech (e.g., different accents and pronunciation styles) and providing accurate transcriptions. By setting the temperature parameter to values between 0 and 1 with increments of 0.1, I intended to explore the trend of results under different temperatures and find the optimal setting that maintains the model's flexibility in recognizing various pronunciations while ensuring accuracy in its transcriptions. This is essential to ensure that the model is both adaptable to different speakers and accurate in its output, supporting the goal of developing an effective automated corrective feedback system.

3.3.4 Collecting and Post-Processing Transcription Data

This subsection explains the process of obtaining the raw transcription data, post-processing the raw text transcriptions and raw log probabilities to get cleaned Whisper's transcriptions and

[7]https://pypi.org/project/openai-whisper/

corresponding confidence scores, and aligning the Whisper-generated transcriptions with the true transcriptions to create the intelligibility classifier with two classification algorithms.

Obtaining Raw Transcription Data. For each input audio file, the Whisper model returns a JSON transcription structured with several keys: 'language', 'duration', 'text', 'segments', etc (see example[8] in the documentation). Specifically, the 'segments' key contains a list of dictionaries detailing information about corresponding segments, including the start and end times of this segment, text (the transcribed text), temperature, average log probability, and more. I used the Whisper English-only model to transcribe the *word-level audio files* with setting language parameter to 'en' (standing for English) to get the transcription data, then extracted and saved the relevant information into a JSON file, where each entry represents the raw transcription data of a *word-level audio file*. One of the entries is shown below:

Example of Raw Transcription Data:

```
's01_bard_1': {
    'studentID': 's01',
    'word': 'bard',
    'trueLabel': 'intelligible',
    'segments': [
        {
            'text': ' Bye!',
            'temperature': 0.1,
            'avg_logprob': -0.7607501029968262,
            'compression_ratio': 0.3333333333333333,
            'no_speech_prob': 0.014807652682065964
        }
    ]
}
```

The raw transcription data consists of 4 keys: 'studentID', 'word', 'trueLabel', and 'segments'. The values of 'studentID', 'word', and 'trueLabel' are derived from the file naming of the preprocessed input audio file. The 'segments' key holds information about the segments provided by the model, encompassing fields like 'text', 'temperature', and 'avg_logprob'. For each segment, Whisper provides an average log probability. To be specific, the target word for this input audio file is 'bard' with the gold annotation 'intelligible'. However, when the temperature parameter is 0.1, the Whisper model transcribed it as 'bye!' which indicates the model thinks this is not the pronunciation for the word 'bard', or in other words, the model thinks this speech data is not intelligible The average log probability for the target word is -0.7607501029968262 which needs to be post-processed into a more human-readable number.

Post-processing Raw Text. In the raw transcription data, the 'text' field includes not only the written word but also a leading space and punctuation marks (e.g., '.', '!', '?'). Additionally, some words may appear in uppercase. To ensure that the Whisper-generated text transcriptions are formatted clearly for comparison with the true transcriptions, a cleaning process of the raw text was conducted. This involves removing the leading space and punctuation marks, and converting all letters to lowercase to ensure consistency and clarity in the transcriptions.

Although the *word-level audio files* are all MP3 files of 1-2 seconds in length, the model may generate more than one segment as the transcription for some audio files. In such cases, all

[8]https://platform.openai.com/docs/api-reference/audio/verbose-json-object

textual contents (values of the ‘text’ field) of the segments are cleaned and then concatenated into a single string.

Post-Processing Raw Average Log Probability for Getting Confidence Score as Threshold. The ‘avg_logprob’ field contains the average log probability of the corresponding segment, providing insight into the model’s confidence. Lower (more negative) average log probabilities indicate lower confidence in the model’s output, while higher (less negative) values indicate higher confidence. To get a more human-readable number as the confidence score that serves as a parameter for analysis, the average log probability needs to be post-processed. In addition, as mentioned there is the case that for a *word-level audio file*, the model may generate more than one segment in the transcription data, leading to two or more average log probabilities provided for only one speech data (one English word). This also makes it necessary to post-processing the raw average log probabilities.

The confidence score was obtained using the following method, which provides a measure of how confident the system is in its transcriptions. Firstly, the logarithmic probabilities (raw values of ‘avg_logprob’) were converted back to linear probabilities (ranging from 0 to 1), inspired by the online discussion[9]. To convert the log probabilities to regular probabilities, the exponential function *math.exp()* in Python package *math* was used[10]. Secondly, the linear probabilities were then rounded to three decimal places for analysis although the exponential function gives result accurate to 11 places. For example, an ‘avg_logprob’ value of 0.7607501029968262 became ‘0.467’ after conversion. This works perfectly for input audio files that received transcription with only one segment since there is only one confidence value for one speech data.

For those input audio files that received transcriptions with more than one segment, initially, I summed up the linear probabilities and divided them by the number of segments to get the confidence score. However, this approach was found inaccurate after calculating the proportion of high linear probabilities. It is not reliable when the model transcribes a single word into several words or even sentences even though it is highly confident in the output. After calculating, there are 122 transcriptions (corresponding to 1345 segments) with multiple segments in the development set, of which 28 had at least one segment with a linear probability exceeding 0.5, accounting for 22.95%. To give a more intuitive understanding, the distribution of all linear probabilities for transcriptions with multiple segments of the development set is shown in Figure 3.1. The linear probabilities are presented on a per-segment basis rather than per-transcription (a transcription can contain several segments, as mentioned in **Post-processing Raw Text** in Subsection 3.3.4). Among the 1345 segments, 703 are with linear probability over 0.5, which is 52.27%.

Obviously, simply averaging the linear probabilities is inappropriate. So the final approach to getting the confidence score for those input audio files that received transcriptions with multiple segments was, to set it to 0.

To apply the confidence threshold in determining intelligibility, an approach of automatic classifying instances was implemented where the confidence score serves as a critical discriminator. This is how it works: first, we define a certain threshold value for the confidence score, and any transcription with a confidence score exceeding this threshold is considered reliable enough to classify the word as ‘intelligible’ or ‘unintelligible’ based on whether the transcription matches the true word. If the confidence score falls below the set threshold, automatically classify the spoken word as ‘intelligible’ without comparing it to the true transcription. This

[9]https://stackoverflow.com/questions/48465737/how-to-convert-log-probability-into-simple-probability
[10]https://docs.python.org/3/library/math.html

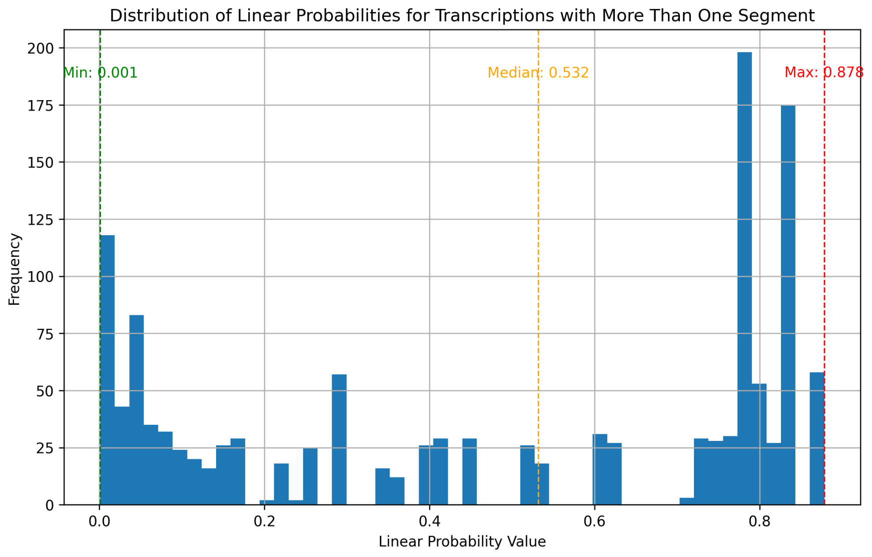


Figure 3.1: Distribution of Linear Probabilities for Transcriptions with Multiple Segments of the Development Set

decision is based on the assumption that when the model's confidence is low, it might struggle with accuracy, and labeling the word as 'unintelligible' could introduce unnecessary bias or error. By doing so, we minimize the risk of mislabeling a correctly pronounced (or intelligible) word as 'unintelligible'.

Confidence thresholds were applied as a way for quality control and error reduction, because setting a threshold allows 'filtering out' transcriptions that the model is less confident about, potentially reducing errors caused by misinterpretations of ambiguous audio. By applying a confidence threshold, I can focus on transcriptions where the model demonstrates higher confidence, thereby reducing the likelihood of incorrect or nonsensical transcriptions being accepted as valid. It allows me to find a balance between correctly transcribing words (intelligibility) and minimizing false positives (mispronunciations or other errors).

The choice of the confidence threshold is crucial, as it impacts the sensitivity and specificity of the intelligibility detection. To explore the impact of confidence scores on the model's performance and find the best threshold to maximize the F0.5 score, a range of threshold values was tested, specifically 0.3, 0.4, 0.5, 0.7, and 0.8. These thresholds were varied in different increments to observe changes in classification metrics and assess how they affect the quality of the model's transcriptions.

3.3.5 Classification Algorithms

After post-processing the texts, the model's predicted labels (referred to as ***Whisper's Judge***) were assigned by comparing Whisper's transcription with the true transcription. If the transcriptions match exactly, the label 'intelligible' is assigned; otherwise, the label 'unintelligible' is assigned.

However, there is a limitation to obtaining labels by simply comparing spellings, which can impact the accuracy of evaluation metrics. For instance, if the model transcribes 'buy' as 'bye', the label assigned by the model would be 'unintelligible' because these words are not identical in spelling. However, they sound the same, and it would be acceptable and more accurate if the model provided a homophone as the written text compared to the true transcription. Another consideration is that the 52 words in the Word List do not occur in a specific context, making this variability acceptable.

To achieve a more accurate analysis, I obtained another type of label (referred to as ***Phonetic Whisper's Judge***). For this purpose, I utilized the phonetic representations obtained from the CMU Pronouncing Dictionary, commonly known as CMUDict[11]. CMUDict is a widely-used public domain lexicon of English words and their phonetic representations, developed by researchers at Carnegie Mellon University (CMU). It provides phonetic representations for over 134,000 words in American English. Each entry in CMUDict consists of a word followed by its corresponding phonetic representations. The phonetic representations are based on the ARPAbet notation, which is a set of symbols designed specifically to represent the sounds of American English using ASCII (American Standard Code for Information Interchange) characters. ARPAbet is not as widely known as other phonetic transcription systems like the International Phonetic Alphabet (IPA), but it is popular in speech recognition and text-to-speech applications because it is computer-readable and uses a limited set of symbols.

Temperature Setting	**Number of Nonsensical Transcriptions**	**Percentage of Nonsensical Transcriptions**
0.0	43	8.33%
0.1	44	8.53%
0.2	44	8.53%
0.3	40	7.75%
0.4	44	8.53%
0.5	51	9.88%
0.6	63	12.2%
0.7	64	12.4%
0.8	87	16.9%
0.9	99	19.2%
1.0	162	31.4%
Test Set: 0.5	246 out of 2115	11.6%

Table 3.4: Percentage of Nonsensical Transcriptions across Different Temperature Settings in the Development Set (516 instances in total)

One thing that needs to be mentioned is that for some texts generated by the model, their phonetic representations are not available in CMUDict, and I assigned a 'N/A' value, indicating 'not available' for phonetic representation. Upon observation of the output, it became evident that these words are not real English words (e.g., 'fundee', 'boltrain', 'whe', 'beginae', etc.), or are misspelled (e.g., 'nows', 'dafft'), or are empty (the model did not provide any transcriptions for the speech data). Similarly, for post-processed texts obtained by concatenating multiple segments into one, where the text contains more than one word, CMUDict cannot provide direct phonetic representations because the phonetic representation corresponds to a single word. Therefore, I also implemented the code to assign a 'N/A' value as the phonetic representation, resulting in a label of 'unintelligible'. This approach is acceptable and does not influence the accuracy because one phonetic representation cannot correspond to multiple words. Taking one word 'farther' as an example from the development data, the model transcribed it as 'right there', which means the phonetic representation would be 'N/A', and

[11]http://www.speech.cs.cmu.edu/cgi-bin/cmudict

thus, an ‘unintelligible’ label would be assigned. Even though the separate words ‘right’ and ‘there’ have their individual phonetic representations in CMUDict (**R AY1 T** and **DH EH1 R** respectively), the combined phonetic representation does not match **F AA1 DH ER0** for ‘farther’ and therefore receives a ‘unintelligible’ label. This type of transcription belongs to model hallucination, which is also a type of nonsensical transcription (will be further explained in Sub-subsection 5.2.3).

The percentage of such nonsensical transcriptions[12] across different temperature values was calculated and is shown above in Table 3.4. The figures suggest that as the temperature value increases, the percentage of nonsensical transcriptions generated also rises. This indicates that higher temperature values lead to a higher incidence of non-sense, misspelled, or non-English words in the model’s output. In addition, the percentage of such nonsensical words in the test set obtained by running the model with the best-performing configuration is also listed in the table, indicating its consistency with the temperature impact.

For comparison purposes, both types of labels were retained: ***Whisper’s Judge*** based on direct text comparison, and ***Phonetic Whisper’s Judge*** based on phonetic comparisons using CMUDict.

The post-processed output of all words with different temperature settings was saved into JSON files. Each entry in these files includes relevant information about the word’s transcription, such as the student ID, the true transcription (word), true label, text transcription generated by the model, text-based labels (*Whisper’s Judge*), phonetic representation obtained from CMUDict for both the true transcription and the model’s text transcription, phonetic-based labels (*Phonetic Whisper’s Judge*), confidence score, and temperature setting used during transcription. This data structure helps detailed analysis and comparison of the model’s performance across varying temperature configurations. Each entry is structured as shown below:

```
‘s01_buy_1’: {
    ‘studentID’: ‘s01’,
    ‘true_transcription’: ‘buy’,
    ‘trueLabel’: ‘intelligible’,
    ‘whisper_transcription’: ‘bye’,
    ‘whisper_judge’: ‘unintelligible’,
    ‘true_phoneme’: ‘B AY1’,
    ‘whisper_phoneme’: ‘B AY1’,
    ‘whisper_judge_phoneme’: ‘intelligible’,
    ‘confidence_score’: 0.467,
    ‘temperature’: 0.0
}
```

In this example, the target word for this input audio file is ‘buy’ with the gold annotation ‘intelligible’ and the model transcribed it as ‘bye’ leading to an ‘unintelligible’ label assigned based on the text comparison. However, homophones ‘buy’ and ‘bye’ both have the same phonetic representation in CMUDict: **B AY1**, which means the first and only vowel sound in the word is a stressed vowel, represented by the numeral 1. This makes the phonetic-based label ‘intelligible’ despite the spelling difference.

[12]All nonsensical transcriptions of the development and test sets can be found in JSON files: *NAs_info.json* and *test_NAs_info.json* respectively.

3.3.6 Evaluation Metrics

In the end, I compared 50 different systems to assess the effectiveness of different configurations, which are the combinations of utilizing five temperature settings (0.1, 0.3, 0.5, 0.7, and 0.9) and five confidence thresholds (0.3, 0.4, 0.5, 0.7, and 0.8) using two classification algorithms (text-based and phonetic-based).

To evaluate and compare the performance of the same model with different configurations for this binary classification task, metrics including *precision*, *recall*, *F1-score*, and *F0.5-score* were used. These metrics are per-class metrics, meaning they were calculated separately for each class ('intelligible' and 'unintelligible'). This approach aligns with macro averaging in that it treats each class independently, without considering its prevalence in the dataset for the metrics. The difference between per-class metrics and macro-averaging metrics is that, in macro-averaging, metrics are calculated for each class individually and then averaged. Moreover, *confusion matrices* were provided to visualize prediction errors and offer deeper insight into the model's performance across different temperature settings.

Precision measures how accurate the predictions are when the instance is correctly labeled as belonging to a particular class. It tells us how well the model does in predicting the correct class. In this project, for both classes 'intelligible' and 'unintelligible', I calculated separate *precision* and *recall* metrics, resulting in two precision values and two recall values. Specifically, ***precision for intelligible class*** answers the question: "Of all the words the model predicted as 'intelligible', how many were truly intelligible?" Similarly, ***precision for unintelligible class*** measures how many predictions labeled as 'unintelligible' were actually mispronounced.

On the other hand, *recall* measures how effectively the model identifies all actual instances of each class. ***Recall for intelligible class*** answers the question: "Out of all the words that are actually 'intelligible', how many did the model correctly predict as 'intelligible'?" The same applies to the 'unintelligible' class, where ***recall for unintelligible class*** indicates the proportion of correctly identified 'unintelligible' words.

The F1-score is the harmonic mean of *precision* and *recall*, providing a single metric that balances these two aspects of the model's performance. Each class will have its own F1-score, reflecting its precision and recall values.

In this project, high precision is prioritized because of the educational context. A pronunciation checker should aim for high precision to ensure that most words marked as 'unintelligible' are genuinely unintelligible, even if it means missing some actual unintelligible words (resulting in lower recall). That is to say, when the model predicts that a spoken word is 'unintelligible' (indicating a pronunciation error), we want to be highly confident that there is indeed an error. This helps to avoid incorrectly labeling correctly pronounced words, which is key to the credibility of the pronunciation checker and prevents learners from receiving incorrect feedback. Thus, for the evaluation of the model's performance, more focus is placed on the ***F0.5-score***, a weighted harmonic mean of precision and recall, with more weight given to precision. This means *F0.5-score* increases the importance of precision while decreasing the emphasis on recall.

Additionally, both text-based and phonetic-based labels were used to evaluate the intelligibility of the spoken data, allowing for a nuanced analysis of the model's ability to handle homophones. This dual evaluation strategy enhances the thoroughness and reliability of assessing the model's performance in this speech-to-task transcription task.

The mathematical formulations of the metrics used in this project are presented as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.3}$$

$$F_{0.5} = \frac{(1 + 0.5^2) * Precision * Recall}{0.5^2 * Precision + Recall} = \frac{1.25 * Precision * Recall}{0.25 * Precision + Recall} \tag{3.4}$$

The evaluation metrics used in this project are derived from four key components: True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). The definition of these metrics changes when focused on a particular class. Here we take the **unintelligible** class as an example to clearly illustrate them:

True Positives are instances where the model correctly predicts a spoken word as 'unintelligible'. For example, if the model predicts a word as 'unintelligible' and the word is indeed unintelligible, this is counted as a True Positive.

False Positives occur when the model misclassifies an intelligible spoken word as 'unintelligible'. This contributes to the model's precision metric since precision for the unintelligible class is about how many of the predicted 'unintelligible' words are indeed unintelligible.

False Negatives are counted when the model predicts a word as 'intelligible' but the word is actually unintelligible. This impacts the recall metric since recall measures how many of the actual unintelligible words are correctly predicted.

True Negatives are instances where the model correctly identifies an intelligible spoken word as 'intelligible'. While TN does not directly affect precision and recall for the 'unintelligible' class, it is a crucial component in understanding the overall performance of the model.

Understanding these components is essential for interpreting the model's precision and recall, as well as the F1-score and F0.5-score, particularly when focusing on the 'unintelligible' class.

Chapter 4

Results

This chapter analyzes the results for both intelligible and unintelligible classes across various system configurations, with a particular focus on the influence of temperature settings and confidence thresholds. The chapter is structured into three main sections. Section 4.1 (Results Across Different Temperature Settings) delves into how varying temperature values impact the model's performance on each class and provides a comparative analysis between the two classes. Section 4.2 (Results Across Different System Configurations) explores the overall trends observed across different combinations of temperature settings, confidence thresholds, and classification algorithms (text-based and phonetic-based methods). This section also identifies the best-performing configuration for running the model on the test set. Finally, Section 4.3 (Results of Test Set under The Optimal Configuration) evaluates the model's performance on the test set using the best-performing configuration, offering insights into its practical applicability and effectiveness in dealing with unseen speech data.

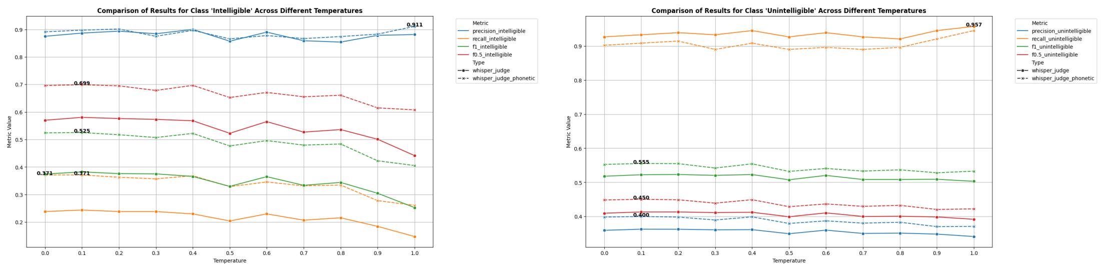
4.1 Results Across Different Temperature Settings

To explore the impact of the temperature parameter, I tested 11 values between 0 and 1 with increments of 0.1 (0 and 1 were also included) on the development data. The corresponding results for both intelligible and unintelligible classes are shown in Figure 4.1.

4.1.1 Results of Intelligible Class

We first look at the results of the intelligible class. It is noticeable that the precision for both the text-based and phonetic-based methods (denoted as whisper_judge and whisper_judge_phonetic respectively in the figure) remains relatively high and stable across all temperature settings, hovering around 0.85 to 0.91. The highest precision 0.911 is achieved at a temperature setting of 1.0 by comparing the phonetic-based labels, which is reasonable since a higher temperature value allows the model to be 'creative' to generate various possible transcriptions for the homophones. Overall, precision is quite consistent across the range, suggesting that temperature settings do not significantly affect the model's ability to correctly predict intelligible words.

However, the recall values are generally low, ranging between 0.15 and 0.25 (text-based method), and between 0.28 and 0.38 (phonetic-based method), both with minor fluctuations, indicating that the model struggles to capture all actual intelligible instances regardless of temperature settings. The highest recall only achieved 0.371 at temperatures 0.0 and 0.1, calculated by comparing the phonetic representation. The low recalls lead to moderate F1-scores, with the highest achieving 0.525 at a temperature of 0.1 also using the phonetic-based method. Conversely, the F0.5-score ranges between 0.45 and 0.7, prioritizing precision over recall. And the highest F0.5-score 0.699 is reached at a temperature of 0.1 for the phonetic-based method.



(a) Results of Intelligible Class (b) Results of Unintelligible Class

Figure 4.1: Comparison of Metrics Results for Both Intelligible and Unintelligible Classes across Different Temperature Settings

4.1.2 Results of Unintelligible Class

The distribution of the metrics results for the unintelligible class shows an exact opposite situation where values of recall even exceed 0.9, followed by F1-scores, F0.5-scores, and precision respectively.

The precision for the unintelligible class in both methods is much lower compared to the intelligible class, ranging from 0.35 to 0.4. This suggests that the model's ability to correctly identify unintelligible words is less reliable. In addition, precision shows minimal changes with temperature adjustments, with the highest precision of 0.400 occurring at a temperature setting of 0.1 for the phonetic-based method. The recall values remain consistently high, ranging from 0.89 to 0.96 for both methods across the temperature range, with the highest recall of 0.957 achieved at a temperature of 1.0 calculated based on the text transcription. The consistently high recall suggests the model's strong ability to capture unintelligible instances regardless of temperature changes, with the exception that the recall increases when temperature is above 0.8.

Due to the low precision, the F0.5-score is relatively low, ranging from 0.39 to the highest 0.450, compared to recall and F1-score. This indicates that improving precision could enhance the overall performance of the unintelligible class. The F1-score for the unintelligible class is moderate driven by the high recall, despite the lower precision, with values between 0.5 and the highest 0.555 (at a temperature of 0.1 for the phonetic-based method).

4.1.3 Comparison between Classes

There is a clear trade-off between precision and recall for both classes, with higher precision correlating with lower recall and vice versa. This trade-off is typical in classification tasks, particularly in scenarios involving imbalanced datasets or nuanced distinctions between classes.

For both classes, temperature settings indeed impact the metrics results, with certain temperatures (e.g., values of 0, 0.1, 1) optimizing different metrics for each class. Many metrics results peaked when the temperature was set to 0.1, indicating that a lower temperature may slightly enhance the performance.

Both the text-based and phonetic-based methods perform similarly in a way that the results of the phonetic-based method are always higher than the text-based method, with the exception

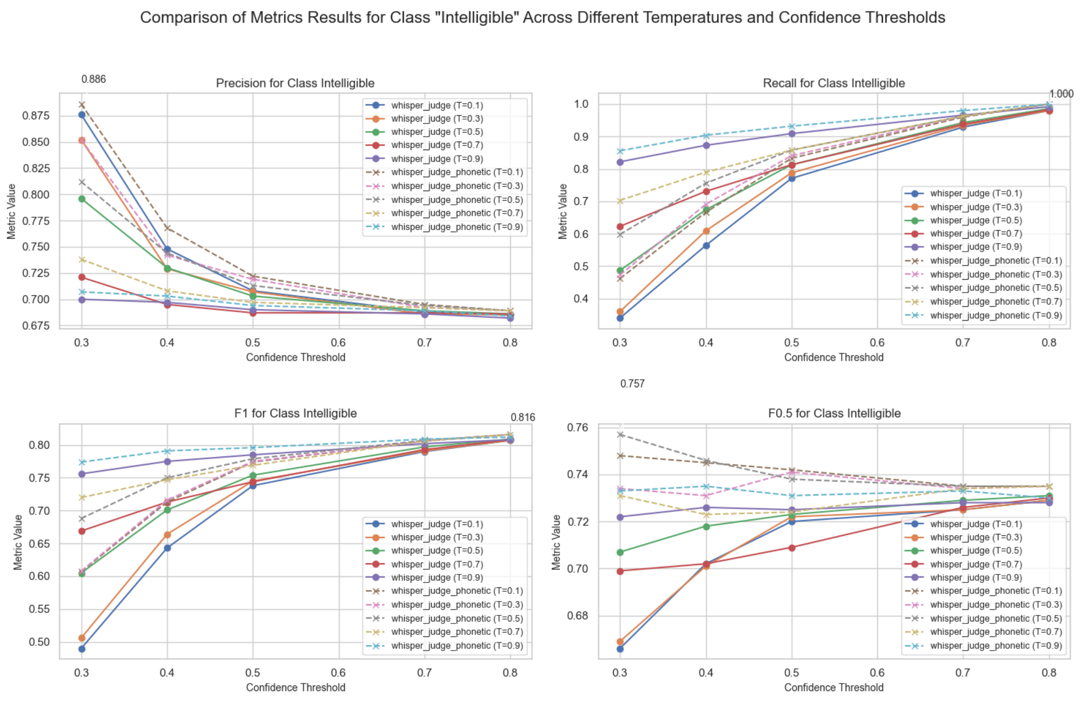


Figure 4.2: Comparison of Results for **Intelligible** Class across Different Configurations of Temperatures (denoted as T) and Confidence Thresholds

that the text-based method offers better recall for the unintelligible class, indicating that the phonetic-based method tends to slightly favor higher precision, while the text-based method offers better recall in most cases.

4.2 Results Across Different System Configurations

After knowing the impact of the temperature settings, instead of using all the 11 values that were tested and reported in Section 4.1, five of them (0.1, 0.3, 0.5, 0.7, and 0.9) were selected to combine with five confidence thresholds (0.3, 0.4, 0.5, 0.7, and 0.8) and the two classification methods to form different 'systems'. In the end, 50 systems were tested on the development set and compared. The results across these system configurations for both intelligible and unintelligible classes are shown in Figure 4.2 and Figure 4.3 respectively.

4.2.1 Overall Trends

To gain insight into how the metrics results vary across different system configurations, we first examine the overall trends of precision, recall, F1-score, and F0.5-score.

For the intelligible class, precision generally decreases as the confidence threshold increases, particularly at lower temperature settings (e.g., T=0.1, T=0.3), indicating that higher confidence thresholds may filter out less precise predictions. In contrast, recall shows an upward trend with increasing confidence thresholds, with higher temperature settings producing better recall values. This suggests that the system becomes more effective at identifying true positive cases (here intelligible instances) as the threshold rises. When examining the F1-score, there is an obvious increase when the confidence threshold increases before reaching 0.5, then

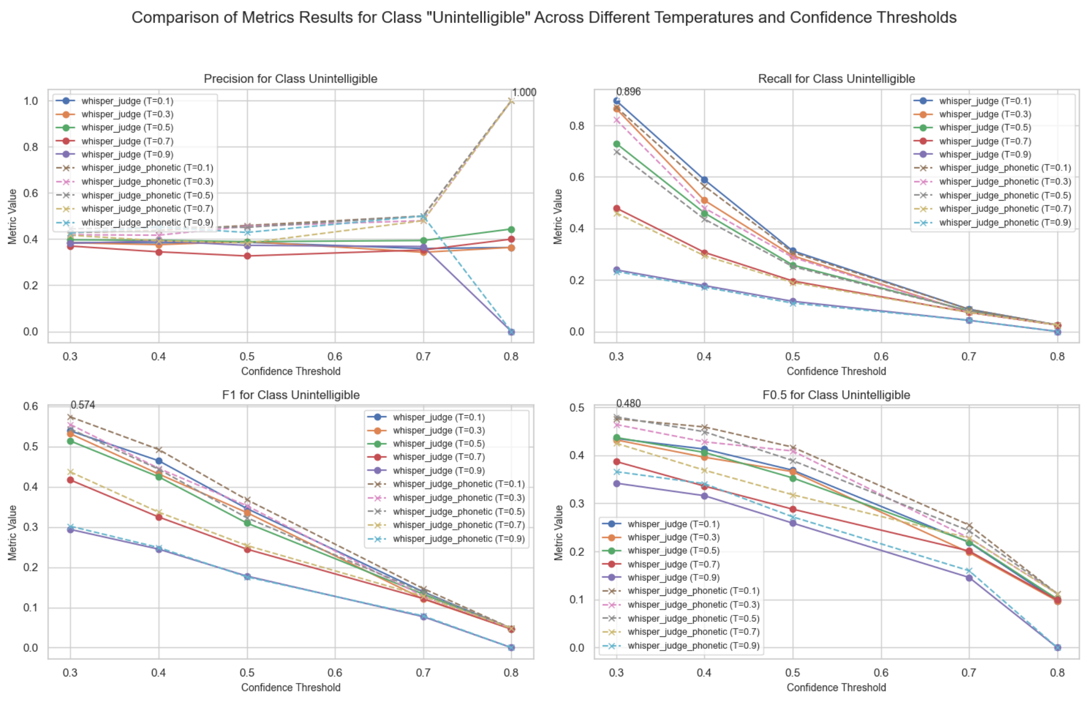


Figure 4.3: Comparison of Results for **Unintelligible** Class across Different Configurations of Temperatures (denoted as T) and Confidence Thresholds

a steady increase to the peak (around 0.8). This peak reflects an optimal balance between precision and recall. Similarly, the F0.5-score for the text-based method generally shows an increase. However, since the F0.5-score places greater emphasis on precision, it leads to a downward trend for the phonetic-based method. Beyond the confidence threshold of 0.7, the F0.5-score stabilizes or slightly declines, reflecting any further improvement in precision becomes smaller or less significant.

On the other hand, the unintelligible class exhibits a different set of trends. The precision for this class across all temperatures remains relatively stable across varying confidence thresholds but experiences a sharp increase at a confidence threshold higher than 0.7 when utilizing the phonetic-based method, with one exception showing a downward trend when the temperature is set to 0.9. In contrast, the recall decreases across all temperature settings as the confidence threshold increases, suggesting that the system becomes more strict and, consequently, more likely to miss unintelligible words at higher confidence thresholds. The F1-score across all temperature settings follows the recall trend, decreasing as the confidence threshold rises, which indicates a reduction in the overall effectiveness of the system in balancing precision and recall at higher confidence thresholds. Similarly, the F0.5-score across all temperature settings also decreases across the confidence thresholds, but more steeply than the F1-score when the threshold is above 0.7, due to its greater emphasis on precision. This sharper decline highlights the challenges of maintaining high precision without sacrificing recall as the confidence threshold becomes more restrictive.

4.2.2 Comparison between Text-Based and Phonetic-Based Algorithms

For the intelligible class, data shown in Figure 4.2 reveals that the phonetic-based method generally outperforms the text-based method across multiple evaluation metrics. In terms of precision, the phonetic-based method shows a slight advantage, particularly at lower confidence thresholds. This suggests that the phonetic-based approach is more effective in accurately identifying intelligible speech data when the confidence level is less strict. Regarding recall, both methods display similar trends, but the phonetic-based method consistently achieves higher recall across all thresholds and temperature settings, indicating its superior ability to capture intelligible instances in the speech dataset. The F1-score is also higher when applying the phonetic-based method, especially at higher temperatures. Similarly, the F0.5-score consistently favors the phonetic-based method, particularly at higher temperatures. This further underscores the phonetic-based method's robustness in prioritizing precision while still achieving commendable recall rates. Overall, the phonetic-based method demonstrates a clear and consistent advantage over the text-based method in handling the intelligible class data, making it a more reliable option for achieving higher accuracy in this classification task.

For the unintelligible class (see Figure 4.3), the precision of the phonetic-based method across different temperature settings and thresholds also shows its advantage over the text-based method. Conversely, regarding recall, the text-based method across all temperatures generally shows higher values at lower thresholds, but this advantage diminishes as the threshold increases. This trend indicates that while the text-based method is effective at capturing unintelligible instances in less strict confidence settings, its performance in recall becomes comparable to or less favorable than the phonetic-based method at higher thresholds. The F1-score across the temperature settings reflects a pattern: the phonetic-based method has an edge at lower thresholds but converges with the text-based method as the threshold increases. This convergence suggests that while the phonetic-based method is initially more balanced in its performance, the difference between the two methods narrows under stricter conditions. Similarly, the F0.5-score across the temperatures shows that the phonetic-based method performs better at lower thresholds but loses its advantage as the threshold rises. This reduction in relative performance at higher thresholds indicates that while the phonetic-based method excels in precision at permissive settings, it does not maintain this advantage under more restrictive conditions.

4.2.3 The Best-Performing Configuration

In this project, different system configurations were tested with the goal of finding the best one that maximizes the F0.5-score for intelligible and unintelligible classes, especially for the unintelligible class. After experimenting, for the intelligible class, the optimal performance with the highest F0.5-score of 0.757 (as shown in Figure 4.2) is observed by using the phonetic-based method with a temperature setting of 0.5 and a confidence threshold of 0.3. Also, the highest F0.5-score for the class unintelligible reached 0.48 (shown in Figure 4.3) when the same configuration as the intelligible class was applied. This best-performing configuration: a combination of temperature=0.5, confidence threshold=0.3, and using the phonetic-based method, was then used to run the model on the test set.

4.3 Results of Test Set under The Optimal Configuration

The results of running the model with the best-performing configuration on the test set are listed in Table 4.1 where the results of the development set under the same configuration are also provided for further comparison. The test set results reveal insights into the performance of the Whisper model for detecting intelligible and unintelligible spoken English words.

For the intelligible class, it achieves a precision of 0.732 and a recall of 0.565, resulting in an F1-score of 0.638 and F0.5-score of 0.691. The precision indicates that 73.2% of the intelligible spoken words were correctly predicted as 'intelligible' by the model. This is a fairly high precision, suggesting that the model is good at avoiding false positives (i.e., not incorrectly labeling unintelligible words as 'intelligible'). The recall is lower than the precision, indicating that the model missed a substantial number of intelligible words, only 56.5% of the intelligible words were correctly identified by the model, leading to a moderate F1-score. However, the F0.5-score gives more weight to precision than recall, reflecting the educational context where false positives are more problematic.

Method	**Metric**	**Intelligible Class**		**Unintelligible Class**	
		Dev	**Test**	**Dev**	**Test**
Phonetic-based	Precision	0.812	0.732	0.445	0.403
	Recall	0.598	0.565	0.699	0.586
	F1-Score	0.688	0.638	0.544	0.477
	F0.5-Score	0.757	0.691	0.480	0.429

Table 4.1: Test Set Results: under the Optimal Configuration (with Development Set Results Provided for Comparison)

For the unintelligible class, the model exhibits a lower precision (0.403) and a slightly higher recall (0.586), compared to the intelligible class. The low precision indicates that it frequently misclassified intelligible speech data as unintelligible, with only 40.3% of the predicted 'unintelligible' words being correct. However, the recall of 0.586 means the model correctly identified 58.6% of the unintelligible words, indicating a decent capability to detect mispronunciation. The F1-score (0.477) reflects a more balanced but lower overall performance for this class. And the F0.5-score of 0.429 emphasizes the model's difficulty in precisely classifying unintelligible words.

From the table, we can observe that the performance of the model on the development set is better than on the test set for both classes. This indicates that the model configuration that performed best on the development set may not fully translate to the test set, as the model shows a slight decrease in all metrics. This slight drop suggests that the model is generally stable and consistent but may still be subject to variability across different datasets or that the test set may contain more challenging examples than the development set, leading to reduced performance.

Chapter 5

Error Analysis

This chapter provides an error analysis for the best-configured model running on the test set, motivated by the need to understand the underlying factors contributing to the model's performance, especially in an educational context where accurate predictions are essential. This chapter contains two sections: Confusion Matrix and Common Types of Errors. Section 5.1 analyzes the distribution of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) to understand the model's performance. Section 5.2 presents some error examples along with their true transcriptions. By dissecting errors that the model encounters, I try to find specific patterns, which can inform future improvements and refinements.

5.1 Confusion Matrix

Understanding where the model succeeds and where it fails not only helps in optimizing it in the future but also ensures that it meets the reliability and accuracy standards required for real-world deployment. To have an intuition behind the best-configured model's performance on the test set, the confusion matrix (5.1) is displayed for inspecting both correct classifications and misclassifications. The first impression from the confusion matrix is the highest number of 797, signifying the correct classification of 797 intelligible instances as 'intelligible', which improves the precision of the intelligible class.

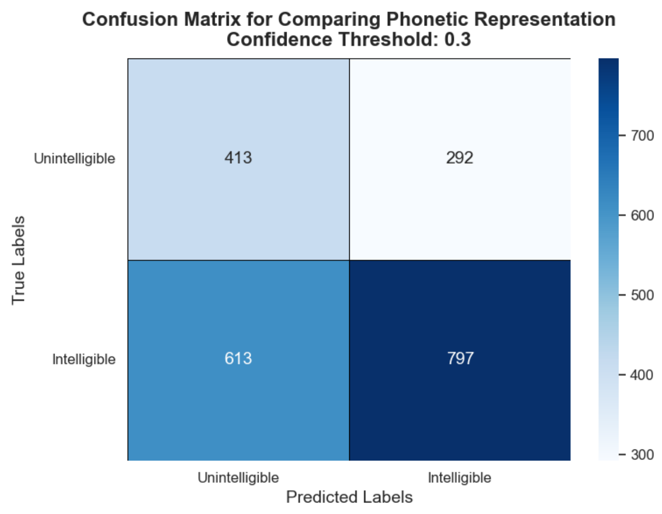


Figure 5.1: Confusion Matrix of the Test Set

However, we analyze with a focus on the **unintelligible** class. Thus, the definition of True Positives, False Positives, False Negatives, and True Negatives changes, as mentioned in Subsection 3.3.6 (Evaluation Metrics).

In this case, the number of **True Positives** is 413, meaning the model successfully labeled 413 words as 'unintelligible' when they were, indeed, unintelligible. This number reflects the model's ability to correctly detect errors in pronunciation or speech clarity, which is essential for applications designed to provide automated corrective feedback. It also influences the recall for the unintelligible class, showing how well the model can identify actual unintelligible instances.

613 is the number of **False Positives**, indicating that 613 intelligible spoken words were misclassified as 'unintelligible'. This high number of false positives indicates that the model tends to be overly cautious, possibly labeling spoken words as 'unintelligible' even when they are correctly pronounced (or, they are intelligible to a human listener). This directly impacts the model's precision for the unintelligible class, as it reduces the percentage of correctly identified unintelligible words out of all words predicted as 'unintelligible'.

For the 292 **False Negatives**, there were 292 cases where the model predicted a spoken word as 'intelligible', but it was actually unintelligible. This shows that the model missed a significant number of unintelligible words, which affects the recall for the unintelligible class. A lower recall means that the model is less effective at catching all errors in speech.

The number of **True Negatives** just mentioned at the beginning, being 797, does not directly influence precision and recall for the unintelligible class, however, it reflects the model's accuracy in affirming intelligible spoken words and shows that the model has a better ability to classify intelligible instances.

5.2 Common Types of Errors

All the errors were extracted and saved to a JSON file[1] for further analysis, classified into false positives or false negatives. This section provides an overview of the common types of errors for both false positives and false negatives observed in the transcription results. Each subsection will delve into specific categories of errors to provide a comprehensive understanding of the issues encountered.

5.2.1 Specific Words Being Misclassified

From the confusion matrix (Figure 5.1) we know that there are 905 miclassifications in total (accounting for 42.8% in the test data), with 613 false positives and 292 false negatives. To find the specific words that were mislabeled, the error frequencies were counted. Table 5.1 provides a quick overview of the top ten frequencies of misclassified words by the model. It shows which words are commonly misinterpreted and whether they tend to be false positives or false negatives.

The table showcases a diverse set of words where the model commonly struggled to differentiate between intelligible and unintelligible pronunciations. Notably, 'bout' had the highest misclassification frequency of 29, with a relatively balanced distribution of 17 false positives and 12 false negatives, indicating a consistent challenge for the model across both error types (false positive and false negative).

Words like 'doll', 'pot', and 'half' predominantly contributed to false positives, suggesting that the model frequently misidentified these intelligible words as 'unintelligible'. Take the

[1]Can be found in the file: *false_positives_negatives.json*

word ‘doll’ as an example, it is counted 25 times as a false positive, meaning this intelligible spoken word was misclassified 25 times as ‘unintelligible’. This is a high frequency of misclassification for this specific word, since for each word, there are 41 or fewer (due to the lack of gold labels or being left-out utterances) audio files, corresponding to 60.1% misclassification rate of the word ‘doll’.

Frequency	Words	False Positives	False Negatives
29	bout	17	12
26	doll	25	1
25	pot, fur	23, 11	2, 14
24	half	24	0
23	paw, pool, pull	13, 18, 6	10, 5, 17
22	pole, pore, seedy, **fir**, **fern**, bird	18, 15, 20, 13, 12, 11	4, 7, 2, 9, 10, 11
21	**tide**, **tied**, bay, daft, pit	14, 13, 18, 18, 9	7, 8, 3, 3, 12
20	cot, caught, putt	18, 10, 9	2, 10, 11
18	pet, boat, pat, bard, pour	11, 16, 13, 13, 13	7, 2, 5, 5, 5
17	pause, dance, board, beer	10, 17, 7, 11	7, 0, 10, 6

Table 5.1: Top 10 Misclassification Frequencies with Corresponding Words

On the other hand, words such as ‘pull’, ‘fur’, and ‘paw’ leaned towards higher false negative rates, indicating a tendency of the model to misclassify these unintelligible pronunciations as ‘intelligible’. This could be a consequence of applying the confidence threshold. There is the case that some of the words are initially correctly labeled as ‘unintelligible’ (being true positives), however, their confidence scores are lower than the threshold, leading to a final label of ‘intelligible’. This confidence threshold impact will be further discussed in Subsection 5.2.2.

The presence of homophones (e.g., ‘tide’ and ‘tied’) and phonetic similarities (e.g., ‘fir’ and ‘fern’), which had high misclassification rates, further underscores the model’s challenges in handling nuanced pronunciation variations.

Overall, the model indeed faced difficulties in distinguishing between intelligible and unintelligible pronunciations for some words, especially those listed in Table 5.1. To better understand why these errors occurred, and try to find if there are any patterns or trends, such as particular phonemes or syllables that the model struggles with, examples of common errors along with their transcriptions are provided, in this subsection with a focus on the false positives.

Transcriptions for False Positives

After manual inspection, some common or interesting errors are provided as examples in Table 5.2. The first column *Word* is the target word, followed by the second column which lists the various incorrect transcriptions generated by the model, with their corresponding frequency counts. In addition, different parts of the transcriptions are highlighted in various colors to better show common errors or patterns in the mistakes.

Notably, the target words ‘boy’ and ‘bay’ are mistranscribed as ‘bye’ with a relatively high frequency, indicating the model tends to think this spoken word is ‘bye’, however, they were actually not. For the word ‘bay’, sometimes the model was able to identify the vowel sound ‘ay’ but struggled to recognize the initial consonant sound ‘b’, generating incorrect transcriptions like ‘may’, ‘they’, and ‘day’, which are near-homophones to the target word ‘bay’. This may happen if the audio quality is poor or if the speaker’s pronunciation is unclear.

Similar situations that occurred are highlighted in blue, and also the words transcribed with different vowels are marked in pink. By observing these blue letters, we can find that the model was confused with some consonant sounds, especially ‘t’ and ‘d’, or ‘t’ and ‘k’. It is interesting that the homophones ‘tide’ and ‘tied’ were frequently mistranscribed as ‘tight’, a near-homophone to the target words, but they got other different transcriptions.

Word	Transcriptions and Corresponding Frequencies
boy (15)	**bye** (14), why (1)
bay (18)	**bye** (8), **m**ay (2), be (2), **b** (2), **th**ey (1), **d**ay (1), main (1), big (1)
bout (17)	**a**bout (9), b**oa**t (4), out (2), go (1), b**u**t (1)
bee (9)	**me** (4), being (3), **d** (1), to (1)
putt (9)	**b**ut (6), part (1), **hutt** (1), bit (1)
mate (9)	ma**k**e (3), ma**d**e (3), meant (1), meet (1), **nate** (1)
tide (14)	tigh**t** (8), time to (1), tides (1), kind (1), fine (1), **h**ide (1), bye (1)
tied (13)	tigh**t** (6), **g**uide (2), sides (1), type (1), kind (1), bye (1), **d**ied (1)
seedy (20)	**cd** (15), city (1), tv (1), see (1), see you later (1), see you then (1)
daft (18)	thats (2), that (2), left (2), **dafft** (2), **daffd** (1), duffed (1), **dufth** (1), **daffk** (1), buffed (1), daphne (1), deaf (1), **ta** (1), adapt (1), **dop it** (1)

Table 5.2: Transcriptions for False Positives

In addition, some transcriptions highlighted in orange are only a letter (e.g., ‘b’, ‘d’) or letters (e.g., ‘cd’). In this case, we are not sure what sounds they stand for, for instance, ‘b’ could be a single consonant sound (/b/ in IPA[2]), or a sound of the letter Bb (/biː/ in IPA). If they are the sounds of how a letter is pronounced, these wrong transcriptions could be seen as near-homophones to the target word. The wrong transcription ‘cd’ for the word ‘seedy’ is a special case. It is not a random combination of letters, instead, it was initially transcribed as ‘CD’, an abbreviation or specifically an initialism of ‘Compact Disc’, which obtained the phonetic representation of **S IY1 D IY1**. ‘CD’ becoming ‘cd’ is due to the post-processing of raw transcription, during which all letters are converted into lowercase. By comparing it with the phonetic representation (**S IY1 D IY0**) of the target word ‘seedy’, we can find that the model actually generated a near-homophone that only the last vowel differs from the target word instead of random letters.

These examples indicate that the model tends to generate near-homophones for the target word when it fails to correctly identify and transcribe intelligible speech data. Despite this error pattern, the model also provides nonsensical transcriptions although this generally occurred rarely in the transcriptions of false positives, with one exception: the target words ‘daft’. The words highlighted in red are nonsensical transcriptions, which are not real English words. It is interesting that for the target word ‘daft’, even though its incorrect transcriptions of ‘dafft’, ‘daffd’, ‘dufth’, and ‘daffk’ are misspelled, their pronunciations are close to the word ‘daft’. This situation also occurred once for words ‘putt’ and ‘mate’, with transcriptions ‘hutt’ and ‘nate’ respectively.

These two error patterns observed in the false positives, namely near-homophones and nonsensical words (or misspelled words) with similar pronunciations, suggest that the Whisper model indeed tends to generate transcriptions that have similar sounds to the expert-labeled intelligible spoken words rather than giving some random texts.

[2]International Phonetic Alphabet

5.2.2 Low Confidence True Positive Instances Turned False Negatives

For the other error type, false negatives (unintelligible words being misclassified as ‘intelligible’), I manually inspected them and found that a substantial of them have the ‘unintelligible’ label, meaning they were actually true positives. However, since their confidence scores are lower than the optimal threshold of 0.3, they were finally identified as ‘intelligible’. The proportion of such words is calculated. Out of all 292 false negatives, there are 244 instances that were initially correctly identified as ‘unintelligible’ by the model but were subsequently relabeled as ‘intelligible’ due to their confidence scores falling below the threshold. This indicates that the confidence threshold plays a crucial role in the model’s performance, potentially causing a considerable number of correct classifications to be overridden and misclassified.

To get an intuition of how confident the model is in these transcriptions, the distribution of confidence scores for these 244 instances is plotted in Figure 5.2. A confidence score of 0 means the transcription was obtained by concatenating the text part of all segments (review how the confidence score was calculated in Subsection 3.3.4).

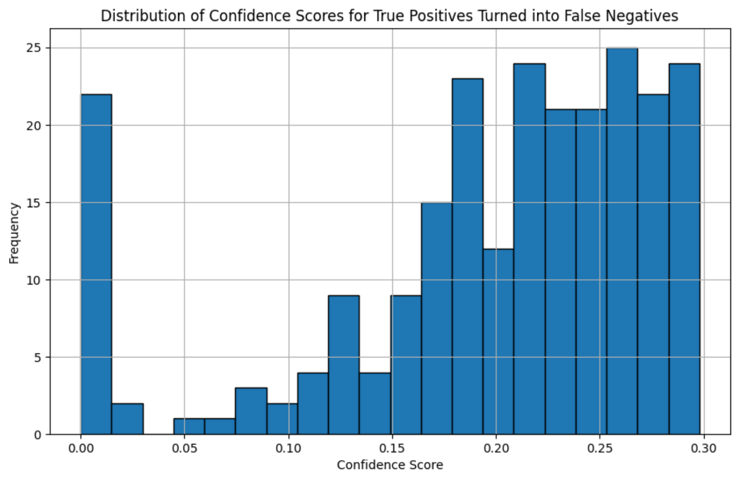


Figure 5.2: Distribution of Confidence Scores for True Positives Turned into False Negatives

From the figure, we know that the transcriptions given have low confidence scores, most of them are within the range of 0.17 to 0.29, indicating the model’s high levels of uncertainty in its output. This also explains why applying a confidence threshold to automatically classify unsure instances as ‘intelligible’ – when the model says ‘there is an error in pronunciation’, we want it to be quite convinced in its judgment to avoid misclassification of intelligible words, which will undermine the credibility of the pronunciation checker.

To better understand how these 244 instances were transcribed, especially whether the model was able to find the specific pronunciation problems, I checked over the transcriptions and compared them with the Excel file[3] where the gold label column is followed by a column that lists the reasons why a word is not intelligible. Some examples are provided in Table 5.3. In this table, the first column is the **unintelligible** target word with the student ID (e.g., s19 meaning student 19), followed by other four columns: the reasons it being labeled as ‘unintelligible’ (named as *Expert-Labeled Feature*), the model’s transcription, the phonetic representations of both the target word and the transcription, and the corresponding confidence score.

[3]See gold labels in: *gold_test_data_with_features.xlsx*

We first observe the top 5 words, which are examples of the model being able to precisely detect the pronunciation errors in phonemes. The first three instances are spoken words that were mispronounced with a wrong vowel sound by MOOC students. Taking the word ‘bout’ as an example, this word was labeled as ‘unintelligible’ by the expert due to the mispronunciation of the middle vowel ‘ou’ (denoted as **AW1** in CMUdict), with the reason ‘different English vowel’. And the model transcribed it as different words ‘boat’ and ‘but’ for different input speech data, which indeed have different English vowel sounds: ‘oa’ (**OW1**) and ‘u’ (**AH1**) respectively from the target word ‘bout’. This example indicates that the model knows there is an error in the vowel sound but actually we are not sure if it gave the correct transcription for the corresponding input audio file since we only know the target word was wrongly pronounced as another vowel, as for which one we are missing the information about it from the gold data.

Target Word	Expert-Labeled Feature	Whisper’s Transcription	Phonetic Representation (Target)	Phonetic Representation (Whisper)	Confidence Score
bout (s19)	different English vowel	boat	B **AW1** T	B **OW1** T	0.212
(s14, s39)		but	B **AW1** T	B **AH1** T	0.297 0.217
putt (s13, s30, s35, s47)	different English vowel	put	P **AH1** T	P **UH1** T	0.239 0.289 0.163 0.281
pull (s30)	long vowel	pool	P **UH1** L	P **UW1** L	0.235
pit (s30, s47)	no aspiration	bit	**P** IH1 T	**B** IH1 T	0.212 0.189
paw (s47)	no aspiration, different English vowel	bow	**P AO1**	**B AW1**	0.155
bout (s11)	different English vowel	boats	B AW1 T	B OW1 T **S**	0.284
meat (s51)	short vowel	mid	M IY1 **T**	M IH1 **D**	0.298
pit (s14)	long vowel	beat	**P** IH1 T	**B** IY1 T	0.268
pit (s19)	long vowel, no aspiration	beach	P IH1 **T**	B IY1 **CH**	0.122
bear (s16)	non-rhotic	there	**B** EH1 R	**DH** EH1 R	0.242
tide (s16)	/d/ is missing	time	T AY1 **D**	T AY1 **M**	0.23
weight (s49)	final consonant missing	why	W **EY1** T	W **AY1**	0.295
bard (s12)	N/A	brought	B **AA1 R D**	B **R AO1 T**	0.225
putt (s17)	different English vowel	thoughts	**P AH1 T**	**TH AO1 T S**	0.277
pat (s19)	no aspiration	**thank you**	P AE1 T	**N/A**	0.219

Table 5.3: Examples of Transcriptions for the 244 FN (Were TP before Applying Threshold)

Similarly, the row for the word ‘pit’ is an example of the model correctly finding the aspiration error, and the row for the word ‘paw’ is an example of successfully detecting various pronunciation errors. It is interesting that for the target word ‘putt’, which occurred 11 times in these 244 transcriptions, the model transcribed it as ‘put’ for 4 different input data (listed in Table 5.3), with different levels of confidence.

However, the cases mentioned are just a rare proportion of the model precisely finding the specific pronunciation errors in the 244 instances. Mostly, the pronunciation errors (e.g., aspiration) found by the model did not match the expert-labeled features, presented as the rest of the examples in the table. The phonetic presentations in red are the wrong reasons the model gave. Taking the word ‘bout’ as an example, although the model successfully found there is an error in the vowel sound, it generated an extra ‘s’ for the nonexistent **S** sound at the end. For words like ‘meat’, ‘pit’, ‘bear’, and ‘tide’, the model successfully detected the errors in the vowel sounds (e.g., mispronunciation of long or short vowels, rhotic sound), however, it made mistakes in the consonant sounds, for instance, transcribing ‘pit’ (s14) as ‘beat’, indicating the model thinks that there is no aspiration for the spoken word’s pronunciation while the aspiration of ‘p’ was made according to the expert-labeled features. The row for the word ‘weight’ is an example of misjudgment of the vowel sound.

Finally, the last three rows are examples of instances where the model generated less similar (or less precise) transcriptions and even unrelated words (e.g., ‘thank you’) for the target words. All these examples demonstrate that, although the model correctly labeled these 244 unintelligible spoken words as ‘unintelligible’, it is not capable of detecting nuanced pronunciation errors in phonemes, which may be an essential property of further providing detailed feedback on pronunciation errors. In addition, the 244 outputs also proved the importance of applying the confidence threshold, which turned all 244 instances into False Negatives, despite the influence of the precision of the unintelligible class.

5.2.3 Nonsensical Transcriptions

As mentioned in Subsection 3.3.5, the model generated nonsensical transcriptions for the development data. This also happened when running the best-configured model on the test data. I mainly divided these nonsensical transcriptions into four categories: model hallucinations, misspelled similar-sounding words, gibberish (not real English words), and empty strings.

Model Hallucinations

‘Hallucination is a response generated by AI that contains false or misleading information presented as fact.’[4] In the context of ASR model, hallucination refers to the model outputting transcriptions that are irrelevant, made-up, or inconsistent with the input speech data. According to Ji et al. (2023) , there are two main types of hallucinations: *Intrinsic Hallucinations* and *Extrinsic Hallucinations*. Intrinsic hallucinations occur when the generated output directly contradicts or misrepresents the source content, while extrinsic hallucinations happen when ‘the output cannot be verified from the source content’.

Based on this definition and analysis of model outputs, it appears that most of the nonsensical transcriptions generated by the Whisper model belong to *Extrinsic Hallucinations*. Specifically, out of all 246[5] nonsensical transcriptions generated for the test data, 166 are extrinsic hallucinations. Several examples of these are provided below:

[4]https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence)

[5]All these instances can be found in the file: *test_NAs_info.json*

- (s13_fern_1) — ‘phone phone phone phone’
- (s13_meat_1) — ‘its really important to see people who see you and to see that i dont know what were talking about we dont know how to do it its a good part of it the other thing is that were going to do that and were going to do that’
- (s16_pool_1) — ‘i dont know what youre doing’
- (s27_paws_1) — ‘who was that’
- (s28_half_1) — ‘have a good day have a good day’
- (s30_seedy_1) — ‘see you later’
- (s35_bout_0) — ‘good job good job good job good job good job good job good job good job good’
- (s39_bear_0) — ‘yeah yeah’

It is interesting that the Whisper model generated some random daily expressions or repeated words or phrases for both intelligible and unintelligible speech data.

Unlike hallucinations in Natural Language Generation (NLG) which have been increasingly studied, the topic of hallucinations in ASR is currently underrepresented(Ji et al., 2023). Hallucinations in speech have mainly been explored for predicting errors (hallucinated word sequences) or identifying errors (Serai et al., 2022; Frieske and Shi, 2024). Koenecke et al. (2024) once studied speech-to-text hallucinations and their occurrence in Whisper, the model used in this project. They discovered that hallucinations are more likely to happen in audio of individuals who exhibit extended periods of non-vocal pauses—a characteristic often associated with aphasia, ‘a language disorder wherein individuals have a lowered ability to express themselves using speech and voice’. However, this cannot explain the reasons for hallucination occurrence since the input data Koenecke et al. (2024) used is at sentence level with an average length of 10 seconds, which is different from word-level input data used in this project.

Among these 166 extrinsic hallucinations, the model generated different hallucinations for the word ‘beer’ 18 times. After listening to audio files of extrinsic hallucinations, I did not find any patterns or reasons why the model had such hallucinations, for instance, the audio files of speakers s27 and s45 are generally of relatively low quality and received 10 and 6 hallucinations respectively, which might be considered a reason of the model generating hallucinations. However, there are 16 audio files made by speaker s49 with good quality (high column, and no background noise) that received hallucinations as well. Thus, the presence of hallucinations in the word-level speech-to-text conversion cannot be solely attributed to audio quality.

Misspelled Similar-Sounding Words

The rest of the 80 nonsensical transcriptions include misspelled similar-sounding words, gibberish, and empty strings. I categorized 50 of them as misspelled similar-sounding words due to their similar sounds of phonemes to the target word. These 50 instances are listed in Table 5.4 where the first column is the target word, followed by a column that provides the transcriptions for speech data made by different speakers. Based on the definition of hallucinations given by Ji et al. (2023), these transcriptions actually can be seen as *Intrinsic Hallucinations* since they misrepresent the source speech data but sometimes are right in representing specific phonemes. However, I categorized them into ‘misspelled similar-sounding words’ because they may show

Target Word	Transcriptions and Corresponding Student Identifiers
bout	bowt (s51)
daft	dopth (s11), duffed (s15, s30), dast (s16), thats (s17, s24), deafs (s19), daffd (s20), fuff (s21), dafft (s23, s46), duht (s26), doft (s27), dufth (s29), guffed (s34), buffed (s38), daffk (s40)
dance	doms (s12), daz (s38)
seedy	zd (s13), cv (s14), cb (s25)
hat	het (s15, s31), hup (s29)
knows	nones (s15), nooves (s39)
bard	bot (s16)
pet	thats (s32, s33)
pause	boz (s35), fows (s43)
nose	nos (s18), nows (s44)
bear	beya (s36), biren (s40)
board	bor (s39)
weight	vate (s39)
pool	booh (s51)
doll	dont (s16, s27, s33, s46, s47)
poor	pua (s21)
bay	beeee (s22)
pull	pooo (s33)
pit	bitro (s35)
pore	borsh (s47)

Table 5.4: Transcriptions for 50 Misspelled Similar-Sounding Words

the model's ability to recognize English sounds to some extent, even though the spellings are wrong and some of the transcriptions are even not real English words.

Interestingly, the word 'daft' is still the one that received the most nonsensical but similar-sounding transcriptions, being consistent with Table 5.2.

Gibberish

I manually selected 21 instances and categorized them as 'gibberish' (shown in Table 5.5) since unlike the 166 extrinsic hallucinations and the 50 misspelled similar-sounding words, they are non-sense and less similar to the sound of the target word. In fact, they can be also seen as *Extrinsic Hallucinations* due to the absence of the nonsensical output in the input speech data. However, they are different from the daily expressions and repeated words or phrases which are real English expressions.

Empty Strings

After counting, there are 9 input audio files the model did not provide any transcriptions for them and returned an empty string, regardless of their intelligibility.

These files are: **s27_paw_1**, **s29_pot_1**, **s29_putt_0**, **s29_put_1**, **s37_pit_0**, **s43_meat_1**, **s43_pit_0**, **s43_putt_0**, and **s50_boot_1**. After listening to these audio files, I found that 5 of them have really low volumes (s29_pot_1, s29_putt_0, s29_put_1, s43_putt_0, and s50_boot_1),

Instance	**Target Word**	**Transcription**
s11_fair_1	fair	fairish
s11_fur_1	fur	fffff
s11_pour_1	pour	pish
s22_farther_1	farther	fargare
s23_pet_1	pet	pyshbegt
s24_paul_1	paul	parlid
s24_pull_0	pull	polloon
s26_half_1	half	hanif
s26_putt_0	putt	boont
s27_put_1	put	boont
s31_put_1	put	puked
s33_bard_0	bard	baudhoo
s40_father_0	father	shother
s40_seedy_0, s44_seedy_0	seedy	3d
s42_farther_0	farther	frustace
s42_pool_0	pool	hmm
s43_city_0	city	cpt
s45_pour_1	pour	bory
s46_pole_0	pole	shulu
s47_pull_0	pull	poudin

Table 5.5: 21 Selected Instances Categorized as 'Gibberish'

which may be the reason the model did not provide transcriptions. However, the rest of the 4 audio files were recorded at normal volume and still received no transcriptions, indicating that there may be other underlying issues or model limitations at play beyond just audio volume.

Chapter 6

Discussion

This chapter delves into the implications of the results presented in the previous chapters (Chapters 4 and 5).

6.1 Discussion of Results

6.1.1 Key Findings

Impact of Temperature Settings and Confidence Thresholds

The study indicates that adjusting the temperature setting of Whisper's English-only base model within a range of 0 to 1 (see Figure 4.1) does not significantly influence the model's precision in predicting the intelligibility of speech data for both intelligible and unintelligible words. Specifically, for the unintelligible class, precision remains stable across different temperature adjustments, while there was a minor fluctuation for the intelligible class across different temperatures but all with high values. Considering the F0.5-score, the overall performance of the intelligible class shows a decline when the temperature is higher than 0.5 due to the deterioration of the recall. This stability of precision for the text-based method across different temperatures is what I did not expect from the model, since according to the documentation, higher values of temperature will lead to more randomness, meaning the possibility of more wrong outputs given. However, the fact is not aligned with my expectation of the temperature impact on precision. For the phonetic-based method, the high precision under high-temperature values is reasonable as the model may generate a broader range of spelling variations for the homophones.

Although the temperature setting does not significantly influence the model's overall performance, it affects the frequency of the model generating nonsensical transcriptions, with a higher rate when temperature increases (see Table 3.4).

The impact of applying a confidence threshold showed a more obvious trend. The model's ability (F0.5-score) to detect pronunciation errors shows a deterioration trend when the confidence threshold becomes more restricted, missing a considerable number of true positives for the unintelligible class. However, in Figure 4.3, there is an exception that the phonetic-based method reached a perfect precision (1.0) of the unintelligible class when it is at a temperature of 0.1, under a high threshold of 0.8. This seems to mean the model's exceptional ability to accurately identify unintelligible speech data when the threshold is set high. However, when we take a look at the corresponding confusion matrix (Figure 6.1), we will find that the high precision of the unintelligible class is at the cost of missing a substantial number of 'unintelligible' instances.

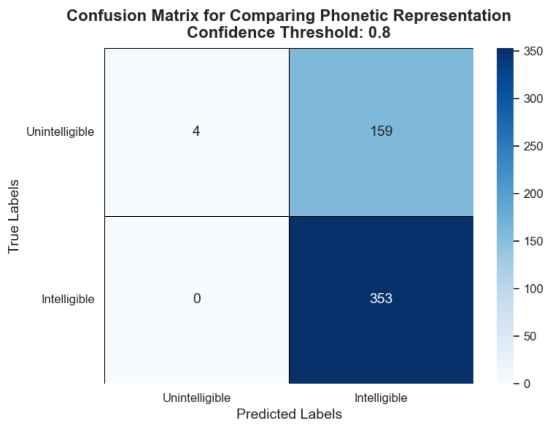


Figure 6.1: Confusion Matrix of High Threshold with Temperature 0.1

This demonstrates that the selection of confidence thresholds is crucial in balancing the trade-off between precision and recall, but its effects were more complex than initially projected.

Robustness of the Phonetic-Based Classification Method

As expected, the phonetic-based method consistently outperformed the text-based method across both datasets. The phonetic-based method's ability to maintain relatively high precision and recall for both classes indicates its robustness and effectiveness in handling varying data conditions, making it a more reliable option for achieving higher accuracy in this classification task.

Tendency to Generate Near-homophones for False Positives

Although the system struggled to give precise transcription for the intelligible instances (ability to correctly classify them as 'intelligible'), it has a tendency to generate near-homophones for them, by distinguishing between similar vowel sounds or phonetic constructs, instead of giving some random texts (without considering model hallucination).

6.1.2 Answering the Research Question

Can Whisper be used for detecting pronunciation problems by checking the intelligibility of speech through a word-level binary classification?

From the overall performance of the best-configured model on the test data on both classes (see Table 4.3), it seems that the model performance is not too bad with F0.5 scores of 0.691 and 0.429 for classes intelligible and unintelligible respectively. However, the model that will be used as a foundation to build an automated corrective feedback system should aim for high values of precision or F0.5-score with relatively high confidence, which can demonstrate its ability to detect errors. However, the transcriptions the model gave are generally with high uncertainty (low confidence score), indicating it is less reliable even though it correctly classifies instances. In addition, the best confidence threshold of 0.3 is already less strict. Under such requirements, the model did not perform as well as I expected. At least in this project, the experiments tested and the optimal model ran on the test data did not show Whisper's English-only base model's advantage in detecting errors for word-level audio files.

6.2 Limitations

The limitations of this study are primarily rooted in the dataset and the inherent constraints of the Whisper model. The dataset used for tuning temperatures and evaluation is in a relatively small size and less representative, which may not fully capture the diversity of English speech, such as variations in accents and background noise. This limitation affects the model's ability to generalize across different speech patterns and may lead to suboptimal performance in real-world applications.

Moreover, the Whisper model itself has certain constraints. As a base model designed and trained primarily for transcribing long-form speech data in 30-second audio chunks (Radford et al., 2023), it is optimized for continuous speech rather than isolated words or very short utterances. This focus on longer segments means that while the model is robust in handling extended speech, its capabilities may be limited when applied to short-form transcription tasks, such as word-level transcription, particularly for audio clips that are only 1-2 seconds long. This mismatch between the model's training environment and the task at hand could contribute to its reduced performance in accurately identifying and classifying brief, isolated spoken words.

6.3 Future Work

6.3.1 Exploring Different Parameter Settings and Multilingual Models

While the current study focused on a specific configuration of temperature and confidence threshold in the Whisper model, future work could explore alternative model settings to potentially enhance performance. One promising direction is the use of the 'best_of' and 'beam_search' parameters. The 'best_of' parameter allows the model to generate multiple transcription candidates and then select the one with the highest probability. This approach could improve accuracy, especially for challenging words where the model might initially produce lower-confidence transcriptions. Similarly, implementing beam search, a decoding algorithm that explores multiple potential transcription paths and selects the most likely sequence could further refine the model's output. Beam search is particularly effective in scenarios where the model needs to disambiguate between similar-sounding words or phrases, which is often the case with unintelligible or mispronounced words.

Moreover, considering that transcription errors in this study were linked to non-native English speakers, future research could also explore the use of multilingual versions of ASR models. Multilingual models are trained on speech data from multiple languages and can better handle variations in pronunciation and accent. This capability could be particularly beneficial when dealing with multilingual speakers, who may have pronunciation patterns that deviate from standard English. Incorporating a multilingual ASR model could improve the detection of mispronunciations and increase the overall robustness of the system.

In summary, experimenting with different model settings, such as 'best_of' and 'beam_search', as well as exploring multilingual ASR models, offers a promising way for enhancing the performance of speech recognition systems in detecting mispronounced words.

6.3.2 Exploring Different ASR Models

Another promising direction for future work involves conducting a comparative analysis between Whisper and other state-of-the-art ASR models. Models like Google's Speech-to-Text, and Microsoft's Azure Speech Service have been widely adopted and are known for their accuracy and scalability. Comparing Whisper against these models could reveal how well it performs in various scenarios, such as recognizing nuanced pronunciation differences or handling background noise. This comparison would help identify the relative strengths and weaknesses of Whisper, and gain deeper insights into its capabilities and limitations in handling unintelligible speech and non-native accents.

Chapter 7

Conclusion

This study aimed to evaluate the effectiveness of the Whisper model, specifically its English-only base version, in detecting pronunciation errors by assessing the intelligibility of speech through word-level binary classification. Through a series of experiments focusing on the impact of temperature settings, confidence thresholds, and two different classification methods, several key insights were uncovered.

Firstly, the influence of temperature settings on the model's precision was less significant than I initially anticipated. While higher temperatures were expected to introduce more variability and potential errors, the model's precision remained surprisingly stable across different temperature configurations. However, a higher temperature did result in a noticeable increase in the generation of nonsensical transcriptions, indicating that while precision may not be drastically affected, the overall reliability of the model could be compromised under such conditions.

The application of confidence thresholds revealed a more predictable trend, where more strict thresholds led to a decline in the model's ability to detect pronunciation errors, particularly for unintelligible speech. This finding underscores the importance of balancing precision and recall in ASR tasks, highlighting the complexity of selecting appropriate model settings.

The study also demonstrated the robustness of the phonetic-based classification method, which consistently outperformed the text-based method in handling varying data conditions. This suggests that phonetic-based approaches may offer a more reliable solution for speech intelligibility tasks, especially in scenarios with diverse or challenging speech inputs.

However, the overall performance of the Whisper model, particularly under the optimal configuration tested, did not fully meet the expectations for accurately detecting pronunciation problems in short, word-level audio files. The model's tendency to generate near-homophones for false positives, rather than random errors, provides some insight into its internal workings, but it also highlights the limitations of using Whisper for this specific task.

In conclusion, while the Whisper model shows promise in certain aspects of speech recognition, its application to word-level intelligibility classification, particularly for detecting pronunciation errors, is limited by its design and training focus on long-form speech. The findings of this study suggest that further exploration of alternative ASR models, as well as the refinement of model settings and methods, is necessary to improve the accuracy and reliability of automated systems for pronunciation assessment.

Bibliography

S. M. Abdou, S. E. Hamid, M. Rashwan, A. Samir, O. Abdel-Hamid, M. Shahin, and W. Nazih. Computer aided pronunciation learning system using speech recognition techniques. In *Ninth International Conference on Spoken Language Processing*, 2006.

K. M. Alhawiti. Natural language processing and its use in education. *International Journal of Advanced Computer Science and Applications*, 5(12), 2014.

E. Bada. Native language influence on the production of english sounds by japanese learners. 2001.

J. E. Flege, O.-S. Bohn, and S. Jang. Effects of experience on non-native speakers' production and perception of english vowels. *Journal of Phonetics*, 25(4):437–470, 1997. ISSN 0095-4470. doi: https://doi.org/10.1006/jpho.1997.0052. URL `https://www.sciencedirect.com/science/article/pii/S0095447097900528`.

R. Frieske and B. E. Shi. Hallucinations in neural automatic speech recognition: Identifying errors and hallucinatory models, 2024. URL `https://arxiv.org/abs/2401.01572`.

Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL `https://doi.org/10.1145/3571730`.

C.-H. Jo, T. Kawahara, S. Doshita, and M. Dantsuji. Automatic pronunciation error detection and guidance for foreign language learning. In *ICSLP*, pages 2639–2642, 1998.

D. Kalikow and J. Swets. Experiments with computer-controlled displays in second-language learning. *IEEE Transactions on Audio and Electroacoustics*, 20(1):23–28, 1972. doi: 10.1109/TAU.1972.1162353.

G. Kawai and K. Hirose. Teaching the pronunciation of japanese double-mora phonemes using speech recognition technology. *Speech Communication*, 30(2-3):131–143, 2000.

A. Koenecke, A. S. G. Choi, K. X. Mei, H. Schellmann, and M. Sloane. Careless whisper: Speech-to-text hallucination harms. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1672–1681, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3658996. URL `https://doi.org/10.1145/3630106.3658996`.

K. Li, X. Qian, and H. Meng. Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):193–207, 2016.

B. Mak, M. Siu, M. Ng, Y.-C. Tam, Y.-C. Chan, K.-W. Chan, K.-Y. Leung, S. Ho, J. Wong, and J. Lo. PLASER: Pronunciation learning via automatic speech recognition. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 23–29, 2003. URL `https://aclanthology.org/W03-0204`.

A. N. Mathew, V. Rohini, and J. Paulose. Nlp-based personal learning assistant for school education. *Int. J. Electr. Comput. Eng*, 11(5):4522–4530, 2021.

X. Qian, H. Meng, and F. Soong. A two-pass framework of mispronunciation detection and diagnosis for computer-aided pronunciation training. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(6):1020–1028, 2016.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

P. M. Rogerson-Revell. Computer-assisted pronunciation training (capt): Current issues and future directions. *RELC Journal*, 52(1):189–205, 2021. doi: 10.1177/0033688220977406. URL `https://doi.org/10.1177/0033688220977406`.

P. Serai, V. Sunder, and E. Fosler-Lussier. Hallucination of speech recognition errors with sequence to sequence learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:890–900, 2022. doi: 10.1109/TASLP.2022.3145313.

H. Strik, K. Truong, F. de Wet, and C. Cucchiarini. Comparing different approaches for automatic pronunciation error detection. *Speech Communication*, 51(10):845–852, 2009. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2009.05.007. URL `https://www.sciencedirect.com/science/article/pii/S0167639309000715`. Spoken Language Technology for Education.

A. Upadhyay, B. K. Sonwani, V. A. Baghel, Y. K. Sinha, A. S. Patel, and M. Ojha. Pronunciation similarity matching using deep learning. In V. K. Gunjan and J. M. Zurada, editors, *Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, pages 305–314, Singapore, 2021. Springer Singapore. ISBN 978-981-15-7234-0.

L. Wang, X. Feng, and H. M. Meng. Automatic generation and pruning of phonetic mispronunciations to support computer-aided pronunciation training. In *Proc. Interspeech 2008*, pages 1729–1732, 2008. doi: 10.21437/Interspeech.2008-466.

Y.-B. Wang and L.-s. Lee. Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):564–579, 2015.

H. A. Younis, N. I. R. Ruhaiyem, W. Ghaban, N. A. Gazem, and M. Nasser. A systematic literature review on the applications of robots and natural language processing in education. *Electronics*, 12(13), 2023. ISSN 2079-9292. doi: 10.3390/electronics12132864. URL `https://www.mdpi.com/2079-9292/12/13/2864`.