Master's Thesis of Arts

Unsupervised Discovery of Non-Categorical L2 Error Patterns Using Wav2Vec2.0 Code Vector Features

Wav2Vec2.0 코드 벡터 특징을 활용한 비지도 방식의 비범주적 L2 발화 패턴 분석

August 2024

Graduate School of Humanities
Seoul National University
Linguistics Major

Eunsoo Hong

Unsupervised Discovery of Non-Categorical L2 Error Patterns Using Wav2Vec2.0 Code Vector Features

Advising Professor, Dr. Minhwa Chung

Submitting a master's thesis of Linguistics

August 2024

Graduate School of Humanities
Seoul National University
Linguistics Major

Eunsoo Hong

Confirming the master's thesis written by
Eunsoo Hong
August 2024

Chair _____(Seal)
Vice Chair _____(Seal)
Examiner _____(Seal)

Abstract

Hong, Eunsoo

Department of Linguistics

Graduate School

Seoul National University

L2 pronunciation is shaped by the interaction of two sound systems, which makes their identity more complex than a single phoneme category. This non-categorical nature demands assessment at a level finer than phonemes. The corresponding sub-segmental inspection, however, is highly labor-intensive, which led to the advent of unsupervised error pattern discovery literature. Nevertheless, previous works fall short of using the supervised and phoneme-prescribed feature, phonetic posterior-gram (PPG) to unsupervisedly discover variation patterns beyond phonemes. Alternatively, this work adopts the Wav2Vec2.0 code vector, a self-supervised learning (SSL) representation acquired through an unsupervised and non-prescriptive process. While maintaining the previous workflow, we aim to understand how well this feature explains sub-segmental variations present in a single segmental error.

To explore the range of variations code vectors capture, we first verify their L1 to L2 discernability via frequency-based usage comparison. From the L1(CMU ARTIC) and L2 (L2 ARCTIC) single-speaker corpora sharing the same reading prompt, probabilities of featural occurrences were constructed into vectors per speaker. These vectors were then clustered to confirm diverging membership. We further dissect within L2 discernment by analyzing patterns among segmentally identical examples. With the model fine-tuned with L1 TIMIT, segmental error detection is run on L2 NIA037 to select error samples submitted for sub-segmental

analysis. For each error type, code vector sequences of corresponding sound frames were extracted from the pre-trained model by referencing the forced-alignment time stamp. We then derived dominant patterns among sequences using the steps of pruning, abstraction, and counting. These patterns are ultimately compared against L1 reference material, likewise created with TIMIT, to interpret their phonetic attribute and relationship with other sub-segmental patterns. Namely, conditional probabilities of phonemes per feature and clustering results among all available raw code vectors in L1 were used for each purpose.

The comparative analysis proved that the code vector usage of L1 and L2 speakers is different with frequency vectors well separated into two clusters on account of nativeness. This difference is marked by the decreasing inventory size in proportion to L2 proficiency, which reflects difficulties in articulating sound units defined in L1 standards. Moreover, sub-segmental patterns possessed the following three common traits that manifested linguistic relevancy, 1) The patterns formed an error continuum along the assumed degree of changed articulatory value, whereby 2) the intermediary typology was ambivalent by assuming opposite values in two codebooks. The gradient positioning highlights the beyond-segmental scope of variation, while the conflicting combination is the literal instantiation of non-categoricity. In line with this trait being an L2 attribute, intermediate patterns were also the least observed in the L1 reference data. Lastly, 3) pattern distribution skewed towards the most approximate sound in the learner's L1, with more foreign targets incurring greater dispersion. This asymmetry shows that variation at its core occurs due to the L1 phonetic transfer. In the end, we claim that code vectors can be an alternative means to evaluate pronunciation gradience, with abilities to quantify the between-categorical position of errors.

Keywords: Pattern Discovery, SSL (self-supervised learning), Code Vector, L2 pronunciation, Non-Categorical, Sub-Segmental, Error Continuum

Student Number: 2022-20727

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Study Background

The idea of non-categoricity in L2 error has been long proposed, as the effect of L1 phonetic transfer. Learners' native speech operates on a different sound system from the target education language. As the two sound systems mutually participate in articulation, they render unique variants that are often hard to strictly categorize through a canonical phoneme grid. In other words, phonetic realizations of L2 span over the boundaries of two or more canonically defined phonemes. This between-categorical characteristic can be best described as existing in a continuum, demanding evaluation in gradience.

In contrast to this need for granular, sub-segmental feedback, current existing technologies in the Computer-Aided Pronunciation Training (CAPT) system primarily rely on segmental judgment. Evaluation metrics utilize phoneme-level mismatch information, namely substitution, deletion, and insertion. Such practice cannot fully consider the gradient nature of errors. The case of substitution, in particular, will run into the problem of misdiagnosis after misrepresenting the between-categorical attribute with a single phoneme category. Cantonese L2 English speakers, for instance, may utter a variation of [n] resembling two L1 phonemes [l] and [n], following their recent sound merger at the syllable initial position (Ng, 2017). Assessing the sound as neither of the approximate phonemes would mirror the true nature of pronunciation (Li et al. 2020). The most direct solution would be to incorporate expert knowledge to sub-categorize the subtleties in acoustic mismatch. While this may amount to the most full-fledged qualitative feedback, sub-segmental expert tagging

entails significant time and labor costs. At best, the supervised discovery of non-categorical error patterns is an impractical route, especially for low-resource languages receiving less attention in academia.

The conflicting need between granularity and practical implementation has been settled by a line of research attempting to automate the sub-segmental error labeling process (Wang and Lee 2013, 2015; Li et al. 2018; Li et al. 2020; Mao et al. 2018). These works expanded the ideas in acoustic pattern discovery to detect non-categorical patterns in the L2 error continuum. Acoustic pattern discovery is a field of research in signal processing that aims to discover recurring signals that can be used as a manual label substitute. By comparing the utterance pair within the data, acoustic sequences of high similarity values are recorded as lexically meaningful units. Likewise, L2 error patterns form repeating signals induced by sound system interaction. Thus, the pattern discovery strategy is well applicable to discovering granular variations present in coarsely labeled phonemes. Previous studies also commonly used phonetic posterior-gram (PPG) generated from MFCC phoneme label-trained neural networks as an examining feature. PPG is a phoneme probability output from the supervised model training, reflecting the final decision boundaries in speech recognition. Thus, the attributes of sound are identified by its association rate with individual phonemes. Ultimately, PPGs were clustered to measure similarity and derive overlapping patterns.

The current research is prompted by the shortcomings of PPGs. For one, they require supervised training of the model to extract the feature from. Studies have shown that the quality and extensiveness of data used for model training affect the reliability of the PPG feature (Li et al. 2020), which would also eventually affect pattern analysis outcomes. With a lack of quality labeled data to train instrumental models, one cannot

be guaranteed a representative description of sound. This partially runs against the purpose of unsupervised error pattern discovery if the goal is to forgo the need for manual annotation. It further implies that the methodology would be hard to apply to L2 speech of low-resourced language that is often short of labeled L1 data in the first place. More fundamentally, however, phonetic posterior-grams are phoneme-circumscribed representations of sound. It is self-contradictory to describe a pattern beyond phonemes with the same tool that confines the judgments that we so long to escape from. What we need instead is a descriptive measure independent of phonemic prescription.

Taking these limitations into account, this paper suggests Self-Supervised Learning (SSL) representation, Wav2Vec2.0 code vector, as an alternative feature to measure non-categoricity. Unlike PPG, representation learning in SSL does not accompany label training. The learned representations are also independent of phonemic classification. After all, SSL is modeled after human cognitive faculty, whereby the audio data alone suffices to acquire languages as an infant (Liu et al. 2022). In the absence of explicit supervised labels, infants self-discover operating units to represent sound, inferring from the data's structural properties. SSL follows this bottom-up fashion of creating pseudo-labels to understand the model input, free from the top-down phonemic stipulation. On top of these comparative advantages, the code vector of Wav2Vec2.0 was chosen for its verified phonetic relevance in model documentation (Baevski et al. 2020; Conneau et al. 2020) and retrievable nature by an entirely model-internal learning process.

Meanwhile, the general idea of adopting SSL representation is conceptually motivated by the procedural resemblance of pattern discovery in representation learning. Acoustic pattern discovery and audio

SSL share the common goal of acquiring a lexical inventory that classifies and expresses speech content. As noted, representations self-discovered by the model are the basic operating units summarizing inherent properties of the input data. In a similar vein, pattern discovery abstracts representational means from structural information of recurring signals. The shared principle further implies that using code vectors is about revisiting the error pattern discovery task with the most updated technology. Above mentioned prior works took place before the recent advancement of End-to-End (E2E) models, whose one-step nature of handling the raw input fostered immense progress in representation learning. While pattern discovery was first introduced to substitute transcription for statistical models in the earlier days, E2E approaches currently lead the SOTA performance, Thus, we may view code vector representation as an advanced rendition of the discovered pattern.

1.2. Purpose of Research

While implementing code vectors for the error pattern discovery task, this work first interrogates the nuanced understanding of L2 variation in the feature. We will then use it to probe the present typologies in segmentally defined substitution errors. Each goal is associated with the following two research questions and corresponding methodological outlines.

Can code vector capture L2 variation differently from the native canonical sound? The sound structural knowledge of self-taught acoustic units extends beyond the familiarity of the data it was trained on. XLSR, the multilingual variant of Wa2Vec2.0, is reported to benefit from cross-lingually generalizable representation learned during the pre-training on distant languages. Simultaneously, the versatility of the representation is

balanced out by inter-language discernability. Frequency probing (Conneau et al. 2020) reveals that the code word distribution was more similar between close languages than languages of different families. In the context of L2 encoding, it is first reasonable to assume that the same set of representations can embody the sounds of L2 while being pre-trained on L1, as the L2 from L1 acoustic divergence is far less than that between different languages. Nevertheless, it is of the question whether they will simultaneously be equipped with nativeness-based discernment. In other words, would the model allocate representations differently for L1 and L2 speech? To verify this, the aforementioned language-similarity probing was extended to within-language L1 to L2 similarity probing.

How are gradient characteristics in pronunciation deviation encoded in code vectors? The previous research question tests the validity of the feature in capturing L2 variation. If the code vector indeed possesses L2 discernibility, how would it capture non-categorical characteristics? What aspects of the feature's attributes are utilized to encode this trait, and can we quantify gradience on a scale reflecting acoustic distance among sub-segmental patterns? Answering these questions was in line with re-evaluating the troubling case of segmental substitution diagnosis. As the goal is to substantialize variations within a single segmental error, we first begin with segmental detection to catalog a list of errors to analyze. From here, temporal sequences pertaining to each error segment were analyzed in an unsupervised manner. Alongside frequency-based similarity probing, core methodologies of previous literature and linguistic probing in the model documentation were implemented. The cluster sequence analysis (CSA) in Li et al. (2018) was first applied to internally derive dominant pattern typologies. To ultimately confirm their separability and attributes, external reference was taken

from L1. Namely, KMeans clustering results of raw code vectors were used to check relationships among discovered sub-segmental patterns, whereas the co-occurrence-based phonetic probing in Baevski et al. (2020) was used for deciphering their acoustic identities. KMeans clustering was a common similarity measure to determine error patterns in Wang & Lee(2013, 2015), Li et al. (2018), and Mao et al. (2018).

The remaining chapters are organized as follows. Chapter 2 will give an overview of the related areas of research. Since this work is inspired by the procedural resemblance of representation learning to pattern discovery, the subtopics are divided into two. The first is the outline of acoustic pattern discovery and its previous application in the field of CAPT. Second, basic principles of audio self-supervised learning will be covered, focusing on the founding criteria of the learned representations. Here, we will introduce Wav2Vec2.0 and its code vector: the SSL model and its representation in usage. Chapter 3 will explain the proposed methodology, centering around the two research questions given earlier. Experimental details on the choice of data and analysis tools will also be expounded. Chapter 4 will lay out the results of the two experiments and summarize their main findings. Subsequently, Chapter 5 will discuss their implications and bring forth other noteworthy observations and how they relate to linguistic understandings of existing literature. Finally, chapter 6 will summarize the main contributions of this work.

Chapter 2. Related Works

2.1. Acoustic Pattern Discovery

Earliest works in acoustic pattern discovery coincide with the task of UTD (Unsupervised Term Discovery). Their objective was to automatically discover words and phrases from raw waveform to complement the zero-resource setting. In the absence of labels, the framework self-discovered present lexical information through repetitive sequences spotted in structural comparison. The comparisons centered around measuring similarities and were either carried out with raw acoustic or model-processed features.

For raw acoustics, MFCC features were primarily used, as was the case for the most foundational literature, Park & Glass (2007). The introduced methodology consisted of two steps. It first calculated the warp path between two utterance-level MFCC sequences using the dynamic time warping (DTW) technique. Within alignment, the goal is to spot areas of lower distortion, which is synonymous with higher correspondence and similarity. Nonetheless, grossly comparing utterances with varying lengths and contents may be inadequate. Thus, a temporal constraint is introduced to limit the global alignment to a local level. Equation 1 stipulates that lengths of pairs participating in alignment cannot differ more than R. Consequently, the distortion can only reach ahead of time by R. It also brings the starting point of alignment under regulation (equation2) to create a natural division of the search grid.

Equation1 $|(i_k - i_1) - (j_k - j_1)| \leq R.$ Equation2 $\begin{matrix} ((2R+1)k+1, 1), & 0 \leq k \leq \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor \\ (1, (2R+1)k+1), & 1 \leq k \leq \left\lfloor \frac{N_y - 1}{2R+1} \right\rfloor . \end{matrix}$

Figure 1 illustrates the temporally constrained local alignments within a

single global alignment pair. This local alignment-obtaining procedure is referred to as segmental DTW. Within each local region, warp paths satisfying length-constrained minimum average (LCMA) are submitted to the second step. The length here is controlled to reflect the desired granularity of discovered patterns. The left graph of Figure 2 illustrates the distribution of selected warp paths across the audio stream.

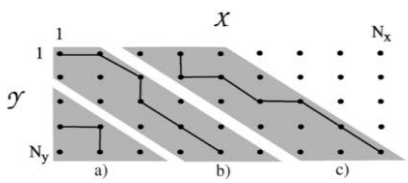


Figure1

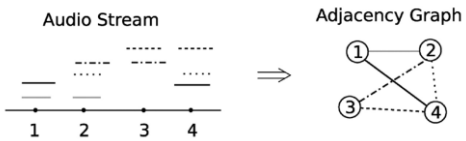


Figure 2

In the second stage, these dispersed entities are united on a similarity basis to ultimately form lexical units. The warp path is first transformed into an adjacency graph that encodes time and distortion information in nodes and edges. Time indices of peak similarity form nodes, while the edge weight denotes the average distortion rate among common alignments. Finally, clustering is performed on an adjacency graph that encompasses all utterances. With all edges initially removed, nodes are iteratively merged in a bottom-up fashion to maximize within-cluster edge weight. The resulting clusters are taken as an inventory of patterns. This is the baseline architecture of unsupervised term discovery, reiterated throughout the follow-up works.

As noted earlier, the main difference in subsequent literature lies in the usage of advanced model-processed features. Since raw acoustic features like MFCC are susceptible to signal variation, intermediary features have been alternatively introduced. These range from acoustic embeddings in a fixed dimension (Kamper et al. 2016) to direct comparison of Gaussian mixture model (GMM) trained per phoneme data (Lee et al. 2015). Most prominently, posterior-grams extracted from

models initially trained with raw features were experimented with. The immediate follow-up work of Park & Glass (2007), for instance, implemented the Gaussian posterior feature as a speaker-robust alternative to the same pattern discovery pipeline. Posterior-gram is a probability vector over the designated model output categories. Namely, the kth dimension of the posterior vector will notate the probability of the output's association with the kth category, which in this case was the Gaussian component. Posterior-gram has been favored for its normalizing property acquired from robust model training. That is, models trained with variable inputs can provide normalization to their processed features. Aside from GMM, other examples of posterior-gram extraction models include Dirichlet Process GMM (DPGMM) (Ravi et al. 2021), Deep Belief Network (DBN) (Lee et al. 2013), Acoustic Phonemic Model (APM) (Mao et al. 2018), MLP (Wang & Lee 2015), DNN (Li et al. 2018), and phonetic segment classifier (Li et al. 2020). The latter 4 models were specifically applied in works of L2 error pattern discovery, as their prescribed categories are phonemic. This means that the feature is mandatorily supervised since categories of probability assignment are predetermined.

In contrast, the number of Gaussian components for GMM posterior can either be prescribed as in Zhang et al. (2010) or be automatically decided using a minimal description length (MDL) principle (Chan et al. 2011). In DPGMM, the number of constitutive clusters is determined through a stick-breaking process, whereby mixing probabilities representing cluster contribution rate is iteratively readjusted. In addition to using DPGMM, Ravi et al. (2021) aggregated the frame-level posteriors to a phoneme level, before subjecting it to a similarity measurement. The phoneme boundaries are determined by referring to the diagonal most faithful to the self-correlation in the affinity matrix.

Subsequently, similarities of phoneme-level posteriors are measured with 3-Neighbor Depth-First Search (3-NDFS) traversal technique which lays out a search route of patterns. While 3-NDFS may differ from adjacency graph clustering, both methodologies are motivated by similarity-based pattern matching. Discussions on phonetic posterior-grams will be deferred to the following section as they specifically concern L2 error pattern discovery.

Besides posterior-grams, Chan et al. (2011) used the model itself to express the signal trajectory of the utterance. Hidden Markov Model (HMM) was constructed per GMM cluster, which took hierarchically clustered MFCC as input. The choice of HMM stems from its capability to model trajectory on top of individual states. Since each GMM cluster represents segments of similar acoustic properties, this feature is referred to as Acoustic Segment HMM (ASHMM). ASHMM behaves as a query, a barometer for comparison in pattern matching. Since it is formed with data that has been clustered twice, it is highly similarity-refined. Thus, although the process does not involve a direct pairwise utterance comparison, the nested similarity measures included in query formation align with the principal methodology of Park & Glass (2007).

2.2 Unsupervised Error Pattern Discovery

As noted, the backbone of the pattern discovery architecture comes down to spotting the recurring signals that can serve as a label substitute. Error patterns of L2 speakers share similar traits with such targets, as pronunciation deviation resulting from L1 phonetic transfer forms repeating patterns with linguistic groundings. Given this parallel, acoustic pattern discovery has been applied to the task of mispronunciation

detection and diagnosis (MDD) in an unsupervised fashion. The unsupervised nature was used to the advantage of identifying granular details that often go unnoticed in simple phoneme-wise detection-based comparisons. However, while all the following works strive for a nuanced understanding, they differ in whether the end goal is a categorical diagnosis or a non-categorical description.

The first four works concern the former, named hereafter as categorical error pattern discovery. The research focus of this thread lies in either constructing a binary decision boundary for evaluation (Lee et al. 2012, 2013) or phonemically decoding an error pattern (Lee et al. 2015, 2016). The two also differ in the locus of comparison.

To begin with, Lee et al. (2012) and Lee et al. (2013) made a cross-speaker comparison with L1 data to derive the peculiarity of non-native speech. The degree of misalignment between native teacher and non-native student utterances is the gauging tool for accuracy evaluation. This misalignment information is derived from the phone-level and word-level features extracted from the relationship matrices. For the phone level, the distance matrix between corresponding L1 and L2 speech was considered. For word level, two self-similarity matrices of the corresponding speech were compared. Features were the estimate of distance and alignment path information in each comparative matrix. Since these linguistic features operate on the acoustic distance, they also would have encompassed a gradient understanding of error patterns. Nevertheless, such information is lost in the face of binary classification training of soft vector machines (SVM). Meanwhile, the initial alignment features differ in Lee et al. (2012) and Lee et al. (2013). The former study uses MFCC and Gaussian posterior-grams, while the follow-up newly introduces the phonetic posterior-gram of DBN to the error pattern

discovery literature. DBN shares foundational traits with self-supervised learning. It follows the pre-training and fine-tuning scheme to minimize the required labeled data. Accordingly, one goal of the experiment was to test how robust mispronunciation detection is along a shift in the amount of labeled data provided during fine-tuning. Experimental results revealed that system performance remained consistent with as little as one-third of the speech annotation. This foreshadows promising results for error pattern discovery mediated by completely unsupervised units in our current work. One difference to be considered for this parallel viewing is that DBN simply learns data distribution layer-by-layer, whereas SSL takes a more foundational approach by learning underlying representations.

On the other hand, Lee et al. (2015) conducted an internal inspection of non-native data to locate patterns unique to each learner. Single-speaker data was analyzed for the size of the variations is a lot tractable at an individual level. They first gathered sound segments for every canonical phoneme referencing the boundaries detected by forced alignment. Next, GMM was trained for each phoneme class, which became the basis of the comparison. As previously noted, these Gaussian models were directly used as features to detect confusable phoneme pairs. The initial global comparison was specified at a local level by reinterpreting segment-level sound as triphones. Frames were divided into three regions that were averaged and then concatenated. Among global confusion pairs that fell below the local distance threshold, patterns were subcategorized according to their triphone identity. Nonetheless, such attempts to derive nuanced sub-segmental understanding were only reduced to segmental interpretation. Error patterns were ultimately incorporated into the Extended Recognition Network (ERN) that used greedy decoding, allowing only single triphone interpretation. The follow-up work ,Lee et al. (2016),

even abandoned using Gaussian posteriors in favor of reverting to the MFCC feature-based DTW approach. Here, the matching process culminated in refining the pattern with ERN-edited forced alignment. More than identifying pattern traits reflecting acoustic distance, these works focused on retrieving the accurate phoneme label to represent them.

The next five works concern uncovering non-categorical patterns and are thus the direct precedent of the current research. Unlike the above-mentioned frameworks, the end goal of these studies is to spot characteristics of L2 pronunciation that cannot be expressed by a defined set of phonemes. This amounts to the task of discovering the unit of expression rather than utilizing the predefined unit for elaborating variation details as in Lee et al. (2015) and Lee et al. (2016). While Wang & Lee (2015) did conduct work on supervised detection on top of unsupervised discovery, classification criteria were sub-phonemic. Using a catalog of patterns belonging to a single canonical phoneme, they constructed a classification pipeline with acoustic models and hierarchal classifiers specialized for each phoneme. Hence, the detection output provided granular information beyond the subsuming phonemic category. This is clearly different from the greedy decoding of ERN that discards the suboptimal variation trajectories. The following non-categorical approaches rather collect these sound routes and analyze them to complete the gradience puzzle. When uncovering these paths in an unsupervised setting, one can either limit the scope of the analysis to a single phoneme or the whole utterance.

At the utterance level, Mao et al. (2018) and Li et al. (2020) used PPGs of the acoustic phonemic model (APM) and phonetic segment classifier to discover non-categorical phonemes. Both works analyzed the shape of the posterior vectors to check how many phonemes a given sound

frame is associated with. If more than one peak was observed among probability distribution, the associated sound was considered non-categorical, with its characteristics defined by peaking phonemes. These non-categorical sounds were notated with subjects of prominent posteriors such as eh_ey and l_n. Ultimately, non-categorical patterns were used to extend the single-category native phoneme set. Mao et al. (2018) initially used state-level posterior-grams of APM that were averaged at a phoneme level. The segment-wise PPG was put through KMeans clustering, whose centroids were the final analyzing component. APM was a deep neural network mutually trained with MFCC and phoneme sequences, including the preceding and succeeding contexts. Because their output was state-based, Li et al. (2020) devised a segmental model to create PPG demanding less intermediary processing. It also took a manual thresholding approach to analyze individual features instead of relying on subsuming analysis per centroid.

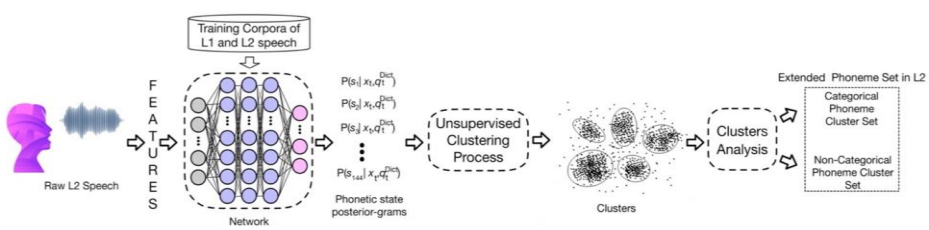


Figure 3 proposed framework of Mao et al. (2018)

Meanwhile, phoneme-based PPG misses the state-level transitory details. To account for variation within the given categorical phoneme, either sounds associated with the same phoneme can be analyzed at a time (Wang and Lee 2013, 2015) or the utterance level analysis can be applied to a segmental scope (Li et al. 2018). In the former case, the restructuring scheme used for fixed dimension mapping in Ravi et al. (2020) and triphone specification in Lee et al. (2015) iterates. First,

frame-level features composing a phoneme segment get hierarchically clustered to form a similarity tree. By thresholding cut-off points in the tree hierarchy, elements of separated subtrees are averaged and concatenated to form a single vector per segment. Segmental feature created in this manner encodes sub-segmental identities. Thus, they are used to discover sub-segmental error patterns using similarity measures of clustering. The clustering algorithms in use were KMeans and GMM with the MDL principle. PPG of these studies was also unique in that it used a mutual phoneme set of multiple languages. The training corpora of MLP, the PPG extraction model, was multilingual. Hence, the posterior vectors were named UPP (Universal Phoneme Posterior-gram).

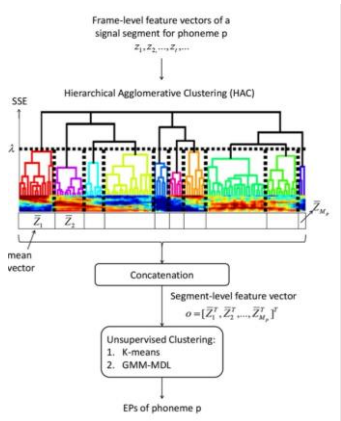


Figure 4 framework of Wang & Lee(2015)

On the other hand, Li et al. (2018) use frame-level clustering of the entire utterances to analyze sub-segmental transitory details. The model for estimating posterior probability was a simple DNN, leading to a frame-wise output. Here, state-based PPGs are directly put through KMeans clustering without the averaging step. Cluster-ID sequences were recorded per phoneme segment, whose boundary was determined by forced alignment. Using the cluster sequence analysis(CSA) algorithm, they selected the most representative patterns out of all observed cluster ID sequences pertained to each phoneme. The current research adopts this

sequence analysis method for analyzing sub-segmental detail represented by code vectors. Thus, the exact details of the CSA algorithm will be further expounded in the methodology chapter.

In the end, existing works in non-categorical pattern discovery commonly use phonetic posterior-gram, a supervised model-processed feature from MFCC raw acoustic input. The models ranged from MLP (Wang & Lee. 2013, 2015), simple DNN (Li et al. 2018) to APM (Mao et al. 2018), producing a frame-level output. Li et al. (2020) constructed a segment-level classifier to reduce the noise of intermediary processing. Methodologies of pattern discovery such as reorganizing sub-segmental realizations and clustering for similarity measure were adopted. Nevertheless, they were not able to entirely escape the categorical circumscription when using phoneme category-based probability vectors. Moreover, the extraction models require supervised labels, which runs against the purpose of making up for L2 annotation deficiency. These PPG limitations were the driving motivation behind this research.

2.3. Audio Self-Supervised Learning

Computational modeling of infant language learning was another long-standing perspective and inspiration behind developing an unsupervised framework. The absence of explicit supervision resembles the learning condition of the newborn, where one has to pick up lexical cues from the statistical property of speech (Park and Glass 2007). Acoustic pattern discovery was, in fact, initially developed by expanding this idea of human cognition. Trust in the self-sufficiency of audio further led to the assumption that models can acquire lexical identities as representational units. Hereafter, audio self-supervised learning was born. Its working hypothesis is that representational units acquired via self-supervision are

generalizable enough to correspond with prescribed labels in the subsequent downstream tasks. That is, knowledge learned performing the preceding upstream tasks may be deduced to match the specificity of the downstream context. Given such universal property, pseudo labels created in upstream tasks have been shown to share relationships with the language-agonistic phonetic conception of sounds i.e., phonemes. With this understanding, this work aims to exploit the quasi-phoneme status of self-taught labels to describe sound, uninhibited by categorizations.

Aside from generalizability, the discriminability of the learned representation matters to guarantee its versatility to various downstream tasks. One way to reinforce the distinctiveness of individual representation is by incorporating contrastive learning objectives. If generalization comes from detecting a correlation between different views of the same object, contrastive learning boils down to learning decorrelation among views of different objects. At its core, the underlying task of SSL is that of a prediction. Representation of the input data is learned while inferring how the model should express probable output. If the locus of the target output is sequential from the last input, the objective is autoregressive predictive coding (APC), while masked predictive coding (MPC) estimates the output of the masked regions. The umbrella terminology encompassing two concepts is contrastive predictive coding (CPC) as it forms the foundation of numerous audio SSL models including Wav2Vec2.0. Accordingly, the algorithms strive for contrast-based prediction, leading to a composite modeling goal that is both generative and discriminative. Generating model prediction is based on the representative knowledge acquired by maximizing similarity between positive samples, whereas discrimination is based on maximizing negative sample distances. If more weight is put on the prediction, the resulting feature will be more faithful to capturing

general traits of the data. Conversely, a higher emphasis on discrimination is more fitting to train features used for manifesting salient outcomes.

A Wav2Vec2.0

Concerning the dual nature of the CPC training objective, featural representations of Wav2Vec can be interpreted as being either more comprehensive or goal-specific. This work posits quantized discrete latent as the former and the context vector as the latter. But before explicating details of features in Wav2Vec2.0, it is important to take an overview of its predecessors.

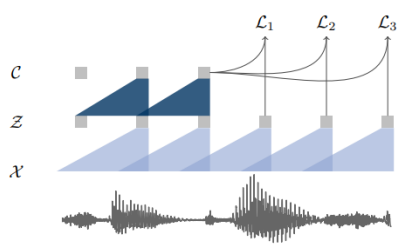


Figure 5 Wav2Vec

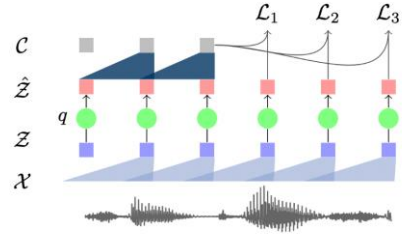


Figure 6 VQ-Wav2Vec

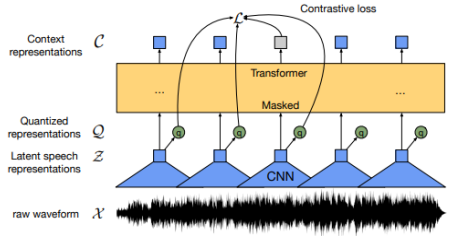


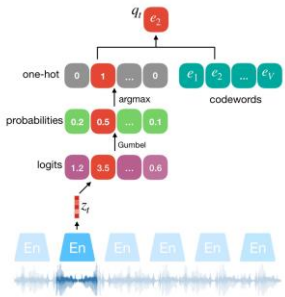
Figure 7 Wav2Vec2.0

We begin with Wav2Vec, the earliest variant of the model. The present architecture is composed of two CNN layers with different specialties. The first layer captures immediate acoustic information, whereas the second layer concerns global context information aggregated up to the present timestamp. As contrastive learning is carried out in an auto-regressive manner, the training objective is APC. Negative samples

are randomly selected from hidden representations of different audio files within the training batch. VQ-Wav2Vec is the next generation of the model and mostly follows the same layout. The sole difference lies in the intervening quantization module between the encoder and context network. The quantization module discretizes the latent representation Z to prep its application for discrete text models such as BERT. This task is either executed with Gumbel SoftMax or online k-means clustering to convert continuous speech waves into discrete one-hot vectors.

Wav2Vec2.0 utilizes the same Gumbel SoftMax tactic. It differs from VQ-Wav2Vec, however, in that the input of the context network does not undergo quantization. It rather uses a continuous context vector trained with a bidirectional MPC objective. Unlike its predecessors, the model predicts the masked portion of latent representations concerning its surrounding context. Quantization is associated with forming the target label for supervision only. Hence, its parameters are fixed during fine-tuning, since the explicit labels are manually provided. From here, we derive the rationale for choosing the discretized code vectors for viewing unprescribed characteristics of sound. While context vectors engage in fine-tuning, they are concerned with substantializing sounds with prescribed labels. In other words, they are goal-specific representations attached to the downstream context. In contrast, quantized vectors are comprehensive representations of sound agnostic to prescribed utilities at the interface. The discreteness also makes it more tractable than continuous context vectors. Moreover, we deemed Wav2Vec2.0 discrete latent as a robust representation with its encouraged efficacy in diversity loss. Another technical advancement of Wav2Vec2.0 is the inclusion of codebook diversity loss, which prevents mode collapse in code word usage. The loss function in Equation 8 shows how diversity loss L_d encourages

the equal usage of every code word. This fair usage complements the contrastive loss L_m that aims to maximize the similarity between context and quantized vector of the same time indices.



$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + n_k)/\tau}$$

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^{G} -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

Figure 8 Gumbel SoftMax Equation 8

The quantized target signal is created by choosing the representative code word entries from multiple codebooks and concatenating them. After multiplying latent features with the quantization matrix, values of the output logits are compared against V code words in G codebooks. The most approximate code word is then selected from each codebook by one-hot encoding the continuous values. Gumbel SoftMax mediates the discrete code words and continuous latent by formulating a probability that is differentiable enough for backpropagation yet can be approximated to the discrete unit with the added noise n and SoftMax transformation. $P_{g,v}$ of Equation 8 denotes this probability of choosing vth code word from gth codebook, given the logit value $l_{g,v}$. Both VQ-Wav-2Vec and Wav2Vec2.0 use 2 codebooks containing 320 code words each. The learning takes place entirely within the model architecture, which makes code vectors retrievable. This contrasts with other SSL models like HuBERT (HSU et al. 2021), where labels from offline clustering are inaccessible.

B Phonetic Relevance of the Codebook

To recap, code vectors are the encoding sound unit of Wav2Vec2.0,

forming the positive and negative samples of the MPC objective. They are created through product quantization that discretizes constitutive sub-vectors with different code words. Under careful regulation of diversity loss, code vectors are proven to bear phonetic relevance. The first receipt comes from the original model documentation of Wav2Vec2.0 (Baevski et al. 2020). The plotted conditional probability of human-annotated phonemes per code vector shows each featural inventory having a specialized phoneme identity. It is worthwhile to note, however, that the reverse does not hold. A single phoneme is not monopolized by a single quantization but is rather represented by multiple varieties. The correspondence asymmetry implies the diverse range of information withheld in the confinement of phonemes. Moreover, it shows that code vectors are more granular units of sound that encode within-phoneme variations. Different vectors corresponding to the same phoneme may each instantiate unique variational aspects. This reaffirms the suitability of code vectors for detailed sub-segmental analysis.

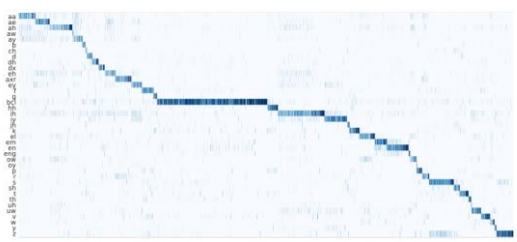
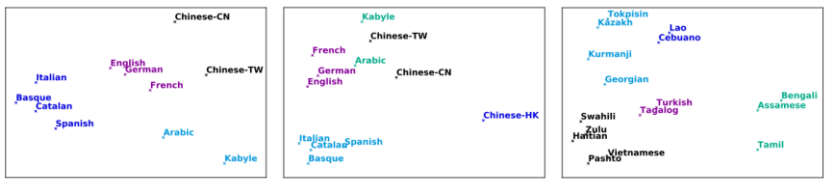


Figure 9 analysis of discrete latent speech representation in Wav2Vec2.0

The next evidence comes from the multilingual pre-trained version of Wav2Vec2.0. In model documentation of XLSR53, code word frequencies among the model's pre-trained language data were investigated (Conneau et al. 2020). Since quantized representations are the acoustic unit discovered from the input training data, it is expected for code words to encode languages of pre-trained data in a linguistically

differentiable manner. Accordingly, usage distributions coincided with language similarity. Figure 10 plots frequency vectors accounting for available discrete tokens (V*G) per language. Vectors of close languages are closer to each other than distant language pairs. Clustering confirms this trend, as the same-colored cluster each represents a language family.



(a) CV-12 model. (b) CV-12 model + zh-HK. (c) BABEL-17 model.

Figure 10 visualization of shared discrete latent speech in XLSR53

Cámbara et al. (2022) is a work that exploits such phonetic relevancy for efficiently initializing a keyword-spotting perceiver. The proposed perceiver architecture performs cross-attention between input data and latent bottleneck at initialization. When codebooks are transferred as weights to this bottleneck (W2V2), the quality of the model improves compared with using a random matrix (BASE). This improvement is fostered by imparted phonetic information from the codebook latent.

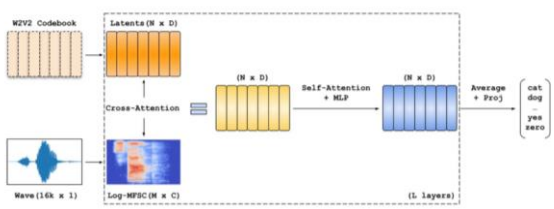


Figure 11 keyword spotting perceiver in Cámbara et al. (2022)

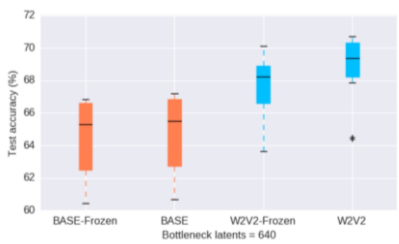


Figure 12 test accuracy of different initialization settings

Chapter 3. Methodology

Figure 13 outlines the overall framework of this research. It is composed of two stages each accounting for previous research questions: 1) Can the code vector capture L2 variation differently from L1? and if so, 2) How may it encode gradient characteristics of error beyond segmental identification? The first stage concerns affirming the eligibility of the featural usage. To use code vectors for describing L2 pronunciations, they must possess L2 discernability, which could be proved through the difference in used inventory between L1 and L2 speech. Thus, the preliminary inventory probing precedes the main research focus on identifying non-categorical patterns. The goal of pattern discovery is to record the variance existing within the same phoneme category when a segmental identification cannot do justice. Such were the cases of misdiagnosis in L2 substitution errors. To concentrate our effort on these troubling examples, conventional MDD methodology was used to compile a list of segmental substitutions submitted for the sub-segmental analysis.

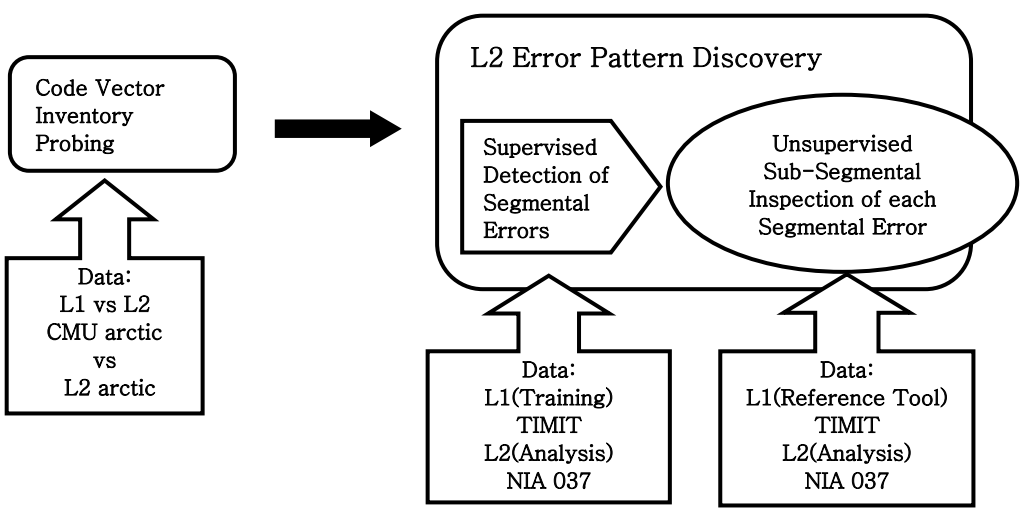


Figure13 overall framework of this research

Segmental detection itself is a supervised process, as it is not the target of our automation goal. It rather represents a prevalent flawed architecture in the current CAPT system, whose shortcomings are set to overcome by unsupervised specification, our real research interest.

3.1 Code Vector Inventory Probing

The first step of our experiment entails validating code vectors as an analyzing tool for non-native pronunciation patterns. The focal point of this endeavor is to prove that the featural usage pattern of L2 speakers differentiates from those of L1 speakers. The final format of product quantization in Wav2Vec2.0 is the concatenation of two entries from each codebook. Thus, code vector usage can be viewed from the perspective of 1) how frequently each entry is used, as well as 2) how frequently each final concatenated form appears. The former view was used in XLSR53 to prove the code vectors' relevancy to language encoding. When calculating the frequency of individual code word entries of the languages the model was pre-trained on, there were distribution overlaps within data of the same language group. We have seen the graphical illustration in Figure 10. The same methodology was adopted with the target of frequency calculation now set as individual L1 and L2 speakers.

For this probing task, comparative corpora of L1 and L2 read speech constructed on the same prompt script was used. Each corpus contained per-speaker recordings over the same reading material. This ensures that the resulting difference in frequency is not the byproduct of the contents of the speech. Using the segmentation information obtained from the forced alignment, we selected the most representative code word entry of the phoneme-divided field in each codebook. Code word entries were retrieved as an index since 384 dimensions of entry size was a less

manageable unit to track patterns. Nevertheless, the raw vectors were not completely out of use. They were later used to identify the characteristics of each code word, as will subsequently be explained in 3.2. The aggregated frequencies of each entry were ultimately transformed into a 1*320-sized vector per codebook. As not all 320 entries were in use, empty slots were filled with the value of 0. Next, frequency values were normalized to a probability from a raw count. Finally, two 1*320-sized frequency vectors were concatenated to perform K-means clustering. The cluster number K was 2, as the expected result was to have L1 and L2 speech data develop separate cluster groups. The separation would reflect the code vector usage differences. Figure 14 gives a graphical illustration of code word frequency retrieval.

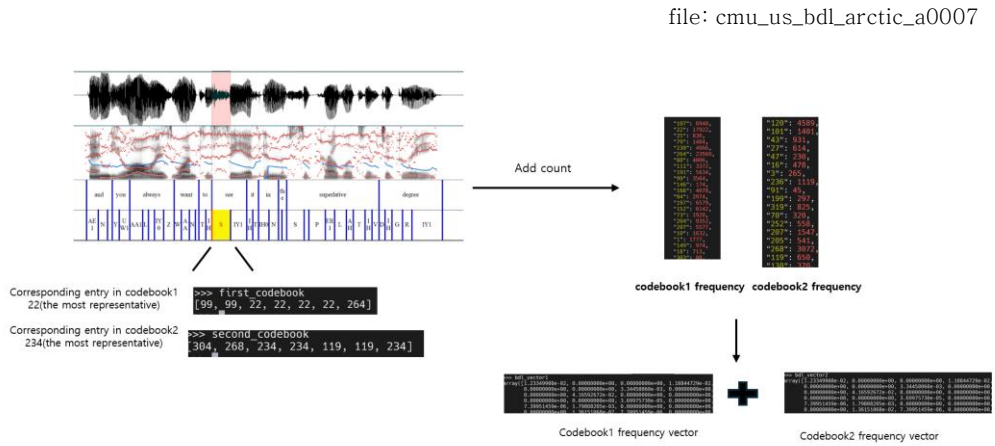


Figure 14 code word frequency retrieval

After calculating the frequency of each code word entry, we took a step further to parse the variations in the representation's final usage format. Probability distributions of separate code words provide only the rough outline of code vector usage as two code words are ultimately paired to represent sound at each time frame. The same frequency retrieval method was applied to a code word pair, counting the frequency

of the concatenated vectors in each speaker's audio data. Unlike the frequency of individual entries, the set of code word pairs employed by each speaker is not fixed. Since each codebook contains 320 code word options, the theoretical maximum of the combined type is 102.4k (Baevski et al. 2020). For the quantizer of the pre-trained model we used, we found that the number varied between 893 to 3133. To plot the frequency in mutual space, we took the union of all available paired indices across the dataset and set them as counting bins to map occurrences. The L1-L2 data we used comprised 5712 different varieties which makes the pair frequency vectors 5712-dimensional. These were likewise normalized into probabilities and clustered into two groups to confirm the deviation of L2 usage patterns from L1. As with the token frequency vector, the code word pairs unused by the individual were allotted the value of 0. The graphical illustration of code word pair frequency retrieval is provided in Figure 15. For comparison, it is demonstrated with the same file as the word retrieval in Figure 14.

file: cmu_us_bdl_arctic_a0007

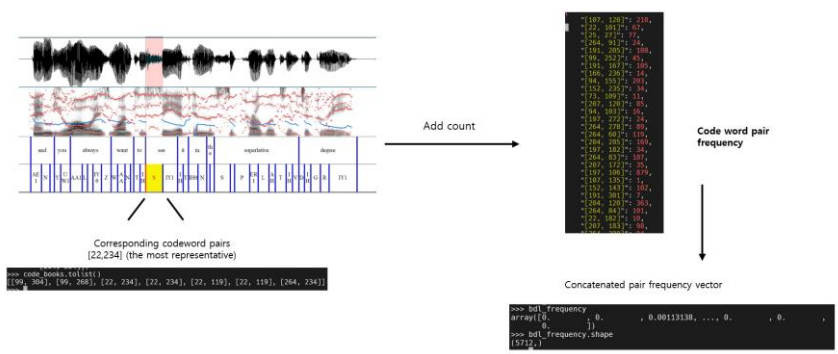


Figure 15 code word pair frequency retrieval

In addition to mass comparison via clustering in vector space, inventory was probed at a more local level through a pairwise comparison between speakers. This need was grounded in the fact that pair

occurrence is a direct reflection of the usage pattern, more so than the individual entry, demanding a more detailed inspection. Accordingly, we have calculated the mutual ratio and count of shared code vector pairs illustrated in Figure 16. The set of employed codeword pairs per individual L1 and L2 speakers was compared against one another to derive a mutual code vector set. This paired set was counted in number (mutual count) and transformed into a probability by dividing it by the total amount used by each speaker (mutual ratio). The mutuality was inspected in terms of both ratio and count because the ratio alone might not bring the full picture into perspective. We have also separately recorded the total number of paired tokens used by each individual and observed a noticeable trend.

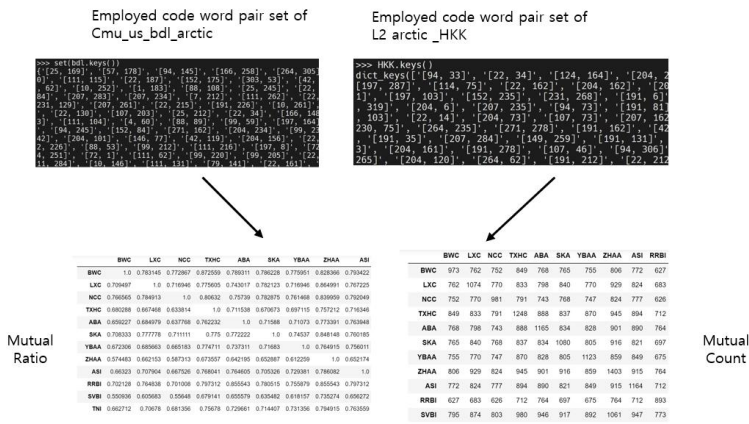


Figure 16 formation of mutual code vector set

3.2 L2 Error Pattern Discovery

After confirming the relationship between code vector representation and speech nativeness, we will embark on the main stage of our research. To achieve the goal of identifying sub-segmental variations, we first need to inventorize the list of substitution errors before dissecting them in an unsupervised manner. This inventorization will be taken as an additional preparatory step, rendering L2 error pattern discovery as a total two-step

procedure. Nevertheless, the prior step is to be taken as an auxiliary task, irrelevant to the goal of automation. It serves no more purpose than limiting the scope of subsegmental analyses as unsupervised discovery is separately applicable to any segment of interest. Fig 17 gives an overview of how the two tasks relate to one another.

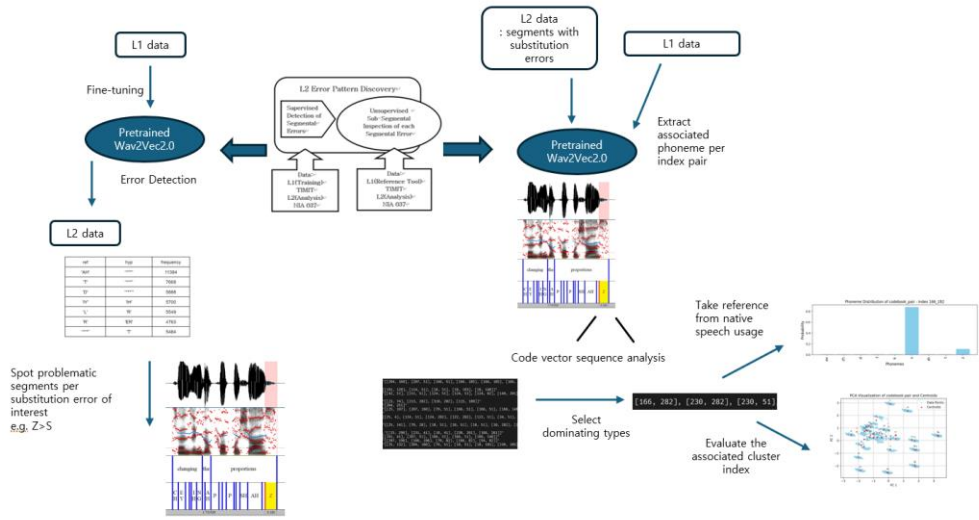


Figure 17 overview of L2 Error Pattern Discovery framework

3.2.1 Supervised Segmental Detection

The left part of Figure 17 narrates the preparatory step of cataloging a list of segmental errors that will be the subject of the subsequent sub-segmental analysis. For segmental error detection, we will use the standard practice in MDD of comparing recognized phonemes with ground truth labels. The recognized phoneme is regarded as how the learner made speech, apart from how it should have been pronounced. To generate phonetic transcriptions of speech for this purpose, the same pre-trained model used for code vector extraction was finetuned with a downstream task in automatic speech recognition.

Then, recognition is run on the target L2 analysis data. After aligning the recognized result with the expected label, the areas of mismatch were recorded and sorted from the highest to the lowest occurrences. As the focus of this research is to overcome limitations in the categorical description of substitution errors, only the cases of substitutions were considered. Other error types, namely insertion and deletion, were excluded. Among the substitution errors recording high frequencies, linguistically relevant patterns prevalent in existing L2 literature were selected. Table 1 shows the example of recorded instances and selected error types annotated with relevant theoretical groundings. Selected substitution errors are italicized.

(*** marks empty presence)

	Ground truth	Recognized result	Frequency	Linguistical grounding
error type: substitution	*Z*	*S*	*11204*	*Lack of voiced fricatives in L1*
error type: deletion	T	***	7668	
error type: substitution	*L*	*R*	*5549*	*a single liquid phoneme in L1 inventory that assume both rhotic and lateral characteristics*
error type: substitution	*AE*	*EH*	*6966*	*Lack of tense-lax distinction in L1*
error type: substitution	AA	AH	4652	
error type: insertion	***	IH	4791	
error type: deletion	N	***	3386	
error type: deletion	*F*	*P*	*2976*	*Lack of labiodental fricative in L1*

Table 1 examples of detected segmental errors

Subsequently, sound segments of the detected errors were retrieved. The retrieval process first involves performing a forced alignment and logging the time stamp of the erroneous sound. The forced alignment results were compared with the recognition result−ground truth alignment, as the start and end index of the falsely recognized phoneme were documented. Figure 18 illustrates the case of retrieving the time stamp of IH to IY substitution error samples. At the upper left is a mismatch instance between the ground truth label and the recognition result. At the lower left, the forced alignment result allows us to document the temporal

index of the erroneous phoneme. In the end, as shown on the right, the goal is to create a list of time stamp information for later usage in code vector extraction. This was viable by using the same phoneme set used during forced alignment when creating labels for the model finetuning data. Since the original L2 data was provided with orthographic transcription only, it had to undergo a grapheme-to-phoneme conversion. The reason is that English orthography has poor correspondence with phonetic realizations, the latter of which is the research interest. To suffice the purpose of both phonemic conversion and notational unification, 39 ARPAbet transcription codes were used.

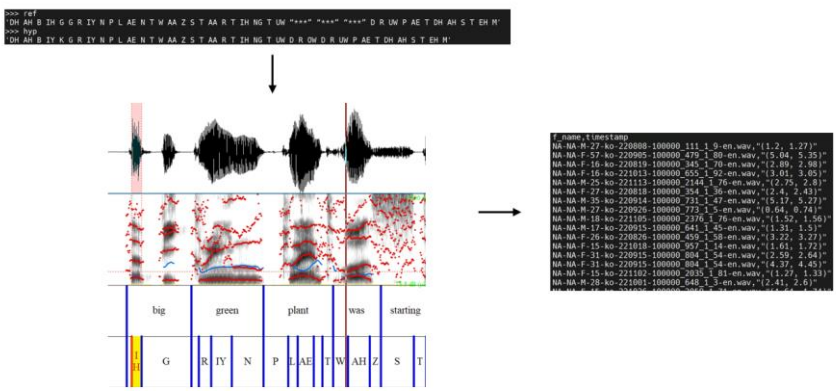


Figure 18 IH to IY substitution sound retrieval

3.2.2 Unsupervised Sub-Segmental Inspection

Once the subjects of non-categorical analysis are laid out, their corresponding code vector encodings are extracted. The extraction was carried out by retrieving the paired-index sequence of the corresponding time frames. To spot recurring patterns and identify the dominant types, cluster sequence analysis (CSA) introduced in PPG-based unsupervised error pattern discovery work (Li et al. 2018) was adopted. The given methodology was previously applied to identify patterns among frame-

wise cluster ID sequences of the segmented phonemes. As this work also attempts to find patterns within a canonically designated segment notated with summarizing indices, this procedure is relevant and applicable. The sequence analysis comprises 3 steps. First, it filters and abstracts the important information by removing ID with a minor presence in the sequence and summing adjacent re-occurrences to one. Second, it selects the dominant types out of the refined representations from the previous stage. Finally, pertinent subsequences are merged with their subsuming counterpart. In all cases, applying the first two steps to code vector sequences resulted in the 3 most dominant types represented by a single paired index. These 3 index pairs were initially taken as a typology of error patterns and were subjected to further analysis. Figure 19 illustrates the error pattern typology discovery process in the case of L to R substitution. We first begin with the sequence analysis method of Li et al. (2018) at the very left. In the middle is the picture of index sequences before deriving prominent patterns. The very right shows an example of how the sequences have been filtered to a final top-three pattern index.

Figure 19 sequence analysis of L to R substitution

While patterns are initially derived through the L2 internal inspection, this may not suffice to identify their attributes and judge their uniqueness. To serve these two aspects of external surveillance, L1 data will likewise undergo code vector extraction to create reference materials. The information to be retrieved from L1 code vector usage is 1) what phoneme each index pair statistically represents and 2) the numerical

values of the raw vectors. The L1 data used to create the reference material is identical to the one used for the finetuning. This is to keep the consistency of evaluation in both segmental and sub-segmental aspects.

The statistics of phonemes at each paired index are, in fact, a revisit to the initial probing experiment performed in Wav2Vec2.0 (Baevski et al. 2020). In the paper, the conditional probability of phoneme distribution per discrete latent is plotted by counting its co-occurrence with human-annotated phoneme boundaries. The same method of counting the co-occurrence is applied, with our L1 reference/finetuning data also being identical to the corpus used here. At each time frame, the encoded paired index is documented alongside the corresponding phonemes. This resulted in a dictionary mapping phonemic occurrences per index. The integer dictionary values are transformed into probability, which is essentially a cross-section of the plotted graph in Baevski et al. (2020) along the vertical axis. This phonemic probing process is demonstrated in Figure 20. Referencing the associated phonemes in L1, we can infer the acoustic attributes of the discovered pattern. If one of the representative index pairs yielded by sequence analysis is [107, 33], for instance, we could guess that the pattern bears the characteristic of nasality and silence.

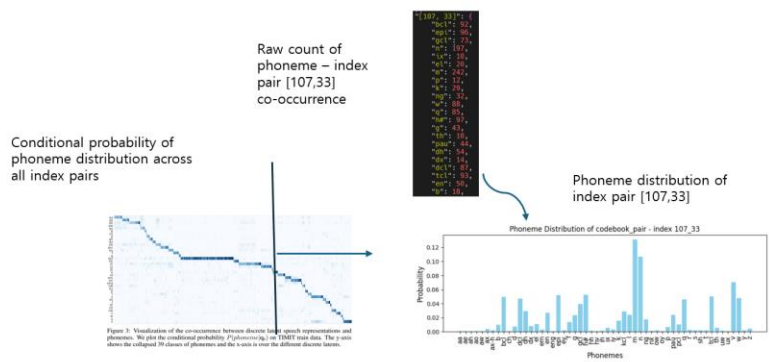


Figure 20 creation of L1 phonemic reference

One notational difference is that while the original Wav2Vec2.0 conditional probing on TIMIT was based on the collapsed 39 phonemes, this research utilizes all 62 phoneme-like-unit (PLU), including the begin and the end marker #h. This reflects the full integrity of annotation to enable more granular judgment when using the distribution plot for reference.

In addition to measuring distribution at the pair level, phoneme distribution per codebook index was also recorded. This was to allow room for analysis in cases where L2 code word pairs are unprecedented combinations in L1. In the former L1 to L2 code vector comparison experiment, we have spotted that index pairs used in L2 often do not coincide with the used set in L1. The only way to decode the attributes of unprecedented pairs will be to take a hint from what each codebook index corresponds to in L1. Figure 21 shows the statistical recordings from each codebook. The [107] index in codebook 1 and [33] index in codebook 2, corresponds to [107, 33] measured in Figure 20.

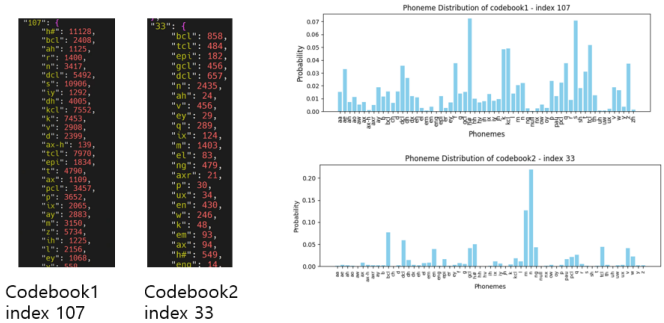


Figure 21 examples of L1 phonemic reference per codebook

Second, to confirm that one pattern marks a clear departure from the other, raw concatenated vectors were mathematically compared in mass. The L1 data utilizes 2202 pairs of code words in total, hence 2202 vectors of 768 dimensions were recorded. These vectors were clustered

into 39 groups, reflecting the 39 ARPAbet phoneme sets used for model finetuning, grapheme-to-phoneme conversion, and eventually segmental error detection. Once three representative patterns were uncovered, their associated cluster ID was retrieved to check if all three belonged to different clusters. If two or more dominant indices turn out to belong to the same cluster, their typological memberships are unified. The 39-cluster division was set as a minimum separability criterion as we wanted to consider patterns that are at least differentiable as the categorization. The clustering result is visualized by applying principal component analysis (PCA) in two dimensions. The visual material makes pairwise distance instantly perceivable. This is important because even if two patterns are separated by cluster ID, the degree of divergence varies. By taking a hint from the visualization and measuring vector-wise Euclidean distance when associated clusters were adjacent, we were able to sort between-pattern relationships from being 1) identical, 2) somewhat similar, to 3) completely different. Figure 22 is the visualization of concatenated vectors, whereby the number denotes the cluster-ID.

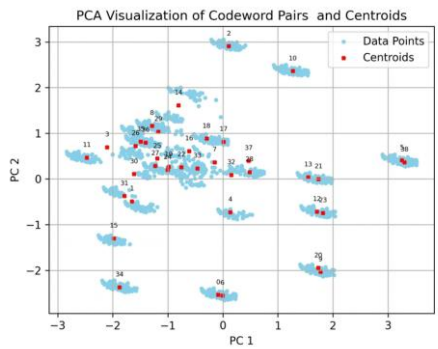


Figure 22 visualization of clustering raw concatenated vectors

For the same analytical purpose as with the phoneme distribution plotting, raw vectors were additionally retrieved at the codebook level. Clustering was not performed on single non-catenated code words as there were not

enough vector types to make grouping into 39 clusters meaningful. The total number of indices used in codebook1 and codebook2 were, in fact, 67 and 215 each. This proves that it is the combination of code words that creates a powerful differentiable latent, thus deserving detailed inspection in 3.1. Fig 23. provides visualization of the non-concatenated forms.

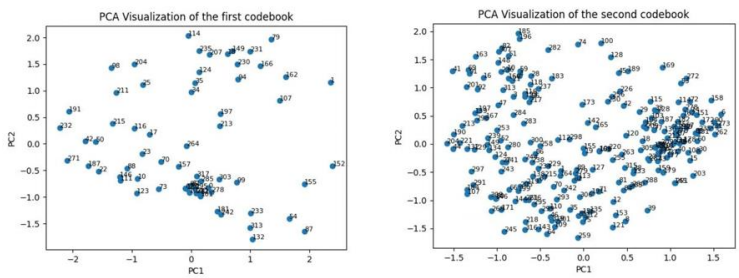


Figure 23 visualization of non-concatenated vectors

In the end, sub-segmental variation analysis can be summarized as the following four steps

1. Retrieve the paired index sequence within the designated time frame associated with each L2 error samples

2. Perform sequence analysis and identify the 3 representative pattern indices

3. Meanwhile, create a referential material with L1 data. One on the corresponding phoneme distribution per index pair / and codebook index, and the other on visualization and numerical recordings of the actual concatenated / and non-concatenated vectors.

4. Use the former to check the attribute of the discovered pattern, and the latter to confirm the number of pattern typologies as well as between-pattern relationships.

3.3 Experimental setting

Dataset: Two different L1-L2 data sets are used for the experiments 3.1 and 3.2. For the inventory usage comparison, it was crucial to choose the read-speech of L1 and L2 with the same prompt script to regulate content-wise variables from interfering with the code vector frequency. Having the data field divided by the individual speaker was also important as the cross-speaker comparison was used to check if there was an overlapping trend differentiating a particular demographic. Accordingly, the chosen corpora are speaker-wise recordings over the same 1132 list of the Arctic reading prompt.

Experiment	3.1. (Inventory probing)	3.2 (L2error pattern discovery)
L1	CMU ARCTIC (5 speakers: bdl, slt, jmk, rms, clb)	TIMIT (segmental detection model fine-tuning: Train) (sub-segmental referential material creation: Train+ Test)
L2	L2 ARCTIC (all 24 speakers of 6 L1 background)	NIA 037 (sentence-wise utterance in train split)

Table 2 overview of the used dataset

For L1, the CMU ARCTIC is a phonetically balanced, US-English single-speaker database primarily designed for speech synthesis research (Kominek et al. 2003). Among 7 speakers in total (bdl, slt, jmk, awb, rms, clb, ksap), 5 speakers of North American accent were used (bdl, rms, jmk, rms, clb). Speakers of other accents such as Indian English or Scottish English were excluded as this research primarily concerns US English. For the L2 counterpart, the L2-ARCTIC encompasses non-native English

data with speakers of 6 L1 backgrounds, Arabic, Mandarin, Korean, Vietnamese, Hindi, and Spanish. Our used version 5.0 includes praat textgrid format of phonetic transcription generated via Montreal Forced Aligner (MFA). As noted earlier, index pair occurrence was calculated by recording the most representative index in the field of the respective phoneme. Thus, this forced alignment information was crucial for frequency calculation. Since CMU ARCTIC did not provide a pre-generated textgrid, we have replicated the L2 ARCTIC generation process using the English (US) ARPA dictionary and acoustic models in MFA.

3.2 had different criteria for corpus selection. For creating reference material, it was most important to have a reliable timestamp to document the corresponding phonemes of each index accurately. Multi-speaker acoustics was also desirable for L1 to fine-tune the model with diverse input to create robust speech recognition. This led us to choose the TIMIT that had manually annotated transcription at the human-confirmed phoneme range. In terms of diversity, TIMIT encompasses 8 dialects of North American regions uttered by a total of 630 speakers.

The same standards were relevant for L2. It was also ideal to have a confirmed phoneme range for the accuracy of the recorded sequence, while the diversity of speakers would allow us to arrive at generalizable findings. Unfortunately, it was hard to find publicly available L2 English data with matching levels of annotation as TIMIT. The read speech portion of NIA 037 Korean English speech data for educational purposes (Han et al. 2024) did meet the second criterion and was chosen. Only the training split was used for the analysis, among which phrase-level utterances were discarded. The reason stems from difficulties in automation. MFA cannot perform phrase-level alignment without manual annotation of each speech chunk, which was impractical to complete as the number of

phrase-length files reached 22101.

Framework: All SSL-related experiments and the model finetuning were carried out under the fairseq[①] framework, a sequence modeling toolkit developed by the Facebook AI Research (FAIR) group.

Pretrained Model: For both code vector extraction and model fine-tuning, an identical variant of the Wav2Vec2.0 model is used. The version used here is the LARGE architecture trained on LibriVox (LV-60k) data. This is also the variant, whereby the original Wav2Vec2.0 discrete latent analysis was performed on. The choice more or less stemmed from the confirmed linguistic relevance here.

Finetuning: Using the TIMIT TRAIN set, the chosen pre-trained model was finetuned under the hyperparameter setting of: CTC criterion, 40000 max update, 3e-4 learning rate, update frequency=4, adam optimizer (betas: 0.9~0.98, eps:1e-08), tri stage learning rate scheduler (ratio 0.1,0.4,0.5), mask probability 0.65, mask channel probability 0.5, mask channel length 64. The final validation phoneme error rate for the finetuned model was 1.63%.

Grapheme to Phoneme Conversion: The phoneme set used for both finetuning labels and forced alignment were the 39 ARPAbet symbols. For finetuning, the set was derived by excluding the sentence stress numeric from the g2p tool kit [②]. For forced alignment, excluding the stress marker in English (US) ARPA dictionary v3.0.0 of Montreal Forced Aligner

① https://github.com/facebookresearch/fairseq
② https://github.com/Kyubyong/g2p

resulted in the same union of symbols. Keeping the representation format identical for model training data and forced alignment was crucial in order to consistently spot the area of mispronunciation designated by the automatic recognition result. These 39 symbols include: *AA, AE, AH, AO, AW, AY, B, CH, D, DH, EH, ER, EY, F, G, HH, IH, IY, JH, K, L, M, N, NG, OW, OY, P, R, S, SH, T, TH, UH, UW, V, W, Y, Z, ZH.*

Forced alignment: As briefly mentioned earlier, the forced alignment tool applied to CMU ARCTIC during experiment 3.1 and NIA037 in experiment 3.2, was Montreal Forced Aligner (MFA). The acoustic model and pronunciation dictionary used for the alignment are English (US) ARPA acoustic model v3.0.0 and English (US) ARPA dictionary v3.0.0.

Clustering and visualization tools: The clustering of paired code vectors and visualization of individual code word vectors were performed with the Facebook AI Similarity Search (FAISS[③])library. This ensured to fasten the process of clustering high-dimensional data on GPU. Visualization of conditional phoneme distribution in 3.2 utilizes the Matplotlib library in Python

③ https://github.com/facebookresearch/faiss

Chapter 4. Results

4.1 L1 to L2 Code Vector Usage Comparison

3.1 explained how L2 from L1 deviation in code vector usage is confirmed at two levels. 1)At the level of individual code word units and 2) a more conclusive format as concatenated pairs. The comparative analysis begins with constructing frequencies of featural occurrences into a vector. These vectors are compared in mass via KMeans clustering with K = 2. Clustering results were additionally evaluated using the Davies Boulding Index (DBI) metric in Equation 3. DBI measures within−cluster scatter (denominator) to between−cluster separation (nominator) ratio. Hence, a lower value marks superior cluster quality. C and σ denote cluster centroids and the average distance from it within each cluster.

$$DBI = \frac{1}{n}\sum_{i=1}^{n}\max_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(C_i,C_j)}\right)$$

Equation 3

In addition to vector clustering, the concatenated pair was granularly inspected via cross−speaker comparison on the used featural inventory.

DBI score: 1.038

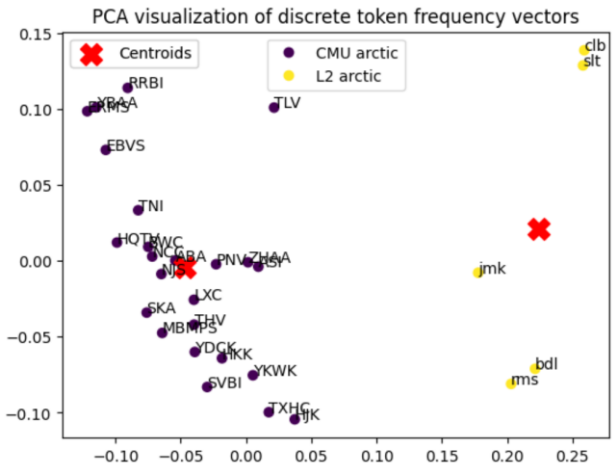


Figure 24 code word frequency vectors clustering result

Figure 24 shows the plotting of individual token frequency vectors across all 29 speakers in L1 and L2 corpora. Although the dimensionality is reduced for visualization, these vectors are originally 640-dimensional. Each dimension represents the frequency of code words that constitute two codebooks of size 320. The speaker of the frequency vectors is annotated next to the plotted location, while each cluster element is color-coordinated according to its cluster membership. As can be noticed from the color scheme, the cluster grouping coincides with the division line between native and non-native speech. Nevertheless, the separation is not the most marked, especially when compared to the inter-language frequency plotting in Figure 10. This is expected since the variation based on nativeness is a lot more subtle than the acoustic distance between different language groups. This motivates us to take a more detailed approach by looking into the frequency of the combined format, beyond individual presence.

DBI score: 0.822

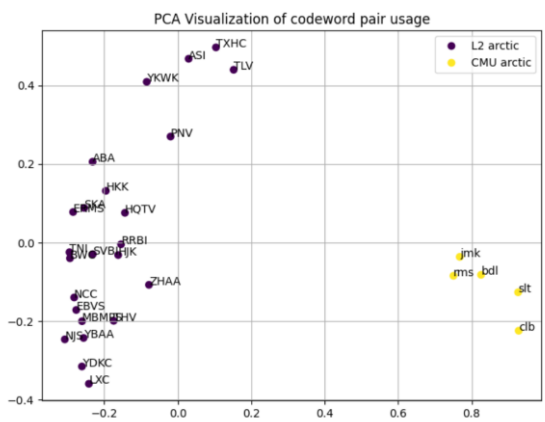


Figure 25 codeword pair frequency vectors clustering result

The paired occurrence, in fact, reveals more abundant information about the difference in code vector usage between L1 and L2. Figure 25 plots the clustering of 5712-dimensional vectors accounting for the

frequency of the union of used pairs among 29 speakers. As mentioned earlier, the amount of total code word pairs employed by each individual varied from 893 to 3133, with a theoretical maximum reaching 102.4k. The number of concatenated varieties rendered by the present data and the model amounts to 5712. In paired clustering, one can spot a clearer separation between the L1 and L2 speaker groups. The DBI score improves when paired vectors are clustered. The paired frequency vector clustering records a score of 0.822, which is superior to the individual usage clustering score of 1.038.

The mass comparison in vector plotting is a rough sketch of how the code vector usage pattern overlaps between speakers. To further illuminate how nativeness affects the encodings of the codebook features, we have conducted a pairwise comparison between speakers. In this regard, Figure 26 first shows the ratio of the total used sets compared to the mutually used overlaps. The number was derived by dividing the length of the mutually used index by the length of the total used pairs of an individual speaker.

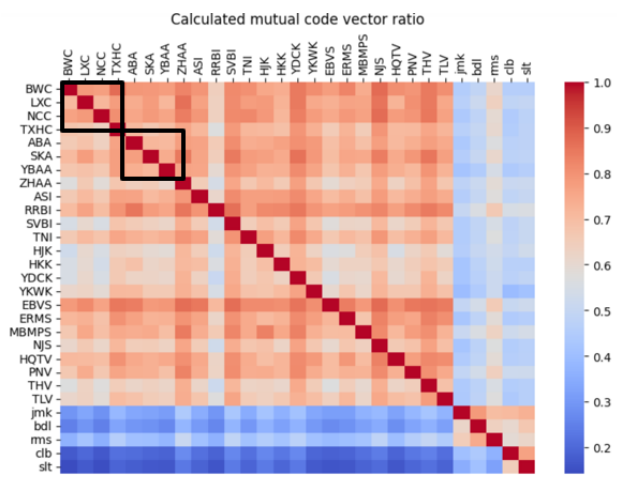


Figure 26

As expected, the sharing rate between native and non-native speakers is far below the rate within the speaker group defined by nativeness. This

reflects the former clustering result in that the used inventory overlaps among each clustered group. Though to a less salient degree, another observable trend is a higher mutuality between L2 speakers of the same L1 background. The above heat map was laid out to have the same L1 background speakers adjacent to each other. Thus, the diagonal region highlighted by the boxed shape shows the mutual ratio between speakers of the same mother tongue. One can notice that the region with a lower degree of mutuality is generally located outside of this diagonal spectrum. Such was a trait that went unnoticed in a macro analysis. This trend is applicable to the L2 arctic speakers from BWC to TLV.

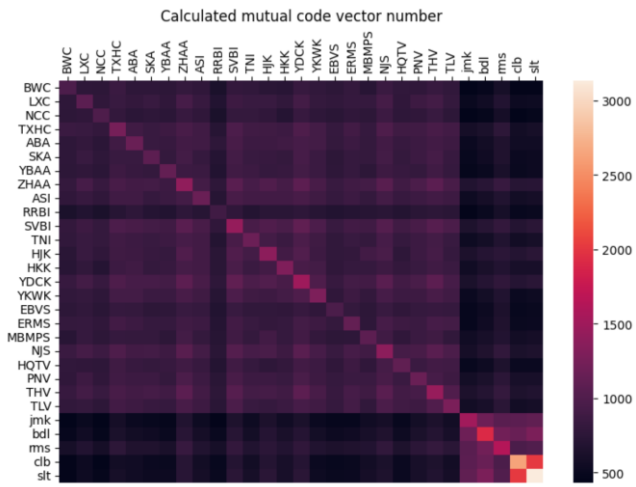


Figure 27

The ratio may not reveal the full picture of the shared aspect. Hence, the raw count of the size of the mutual set was additionally considered. The unfiltered number, indeed, revealed several additional information. For one, the number of code vector pairs utilized by the native speaker was far beyond that of non-native L2 speakers. In Figure 27, the diagonal intersection of the left side has darker shades associated with a smaller number of tokens compared to the rightmost five grids of CMU ARCTIC speakers.

Meanwhile, the used vector size among L2 speakers on the left

varies. The question followed whether there was an underlying factor determining how many index pairs one would use for their non-native speech. Observing the individual recording of the raw count, we noticed that the size increased proportionately to the speaker's language proficiency. Table 3 is taken from the original corpus documentation (Zhao et al. 2018). Demographic information of speakers present at the data's initial release is provided. Based on the TOEFL iBT score, we selected two speaker groups representing opposing levels of speaking proficiency. The lower-level group consists of speakers SKA, EBVS, BWC, and LXC, whose scores range under 90. The higher-level group consists of speakers HKK, YDCK, and NJS, whose scores are equal to or greater than 110. Table 4 is the numerical recordings of the mutual code vector count. One can notice that higher-level speakers have more overlap with 5 native speakers compared to their lower-level counterparts. Hence, even when the mutual ratios did not mark a clear difference in the heatmap, the absolute number of shared tokens varied as a function of articulation adeptness.

Table 3 demographic information of L2 arctic speakers

Speaker	L1	Gender	TOEFL iBT
HKK	Korean	M	114
YDCK	Korean	F	110
BWC	Mandarin	M	80
LXC	Mandarin	F	86
YBAA	Arabic	M	100
SKA	Arabic	F	79
EBVS	Spanish	M	70
NJS	Spanish	F	110
RRBI	Hindi	M	91
TNI	Hindi	F	99

Less proficient

	jmk	bdl	rms	clb	slt
BWC	440	476	597	447	446
LXC	501	555	683	521	531
EBVS	479	520	634	489	505
SKA	491	542	657	513	520
HKK	550	617	685	604	607
YDCK	627	696	771	702	728
NJS	584	660	789	625	645

More proficient

Table 4 mutual code vector count among different proficiency groups

Table 5 shows that the variation in overlapping degree further translates into the overall amount of used paired tokens. One possible reason behind this phenomenon is that to fully utilize the self-supervised learning

feature encoding English, one has to be phonetically aware of the sounds in the language. This awareness is marked by the range of utilization ratio of the code vector inventory. If the amount used by the native speakers is the full range of available acoustics, the less proficient one is at articulating acoustic units defined in the L1 standard, the less amount of inventory in use there will be. The language here does not particularly pertain to English but rather applies to any variant used during pretraining, whereby quantization weight is learned.

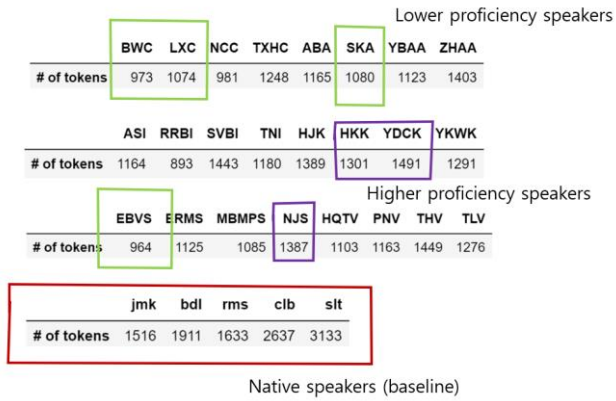


Table 5 number of code vectors utilized by each speaker

4.2 L2 Error Pattern Discovery Result

The L2 error pattern discovery begins with selecting a list of substitution errors that becomes the subject of sub-segmental analysis. Table 1 briefly mentions several examples of the chosen substitution errors. 4.2.1 enumerates all chosen instances, with a linguistically relevant reason behind the selections.

4.2.1 Detected Segmental Errors

The types of phonetic sounds involved in the substitution errors are divided into three categories: fricatives, liquid, and vowels. As recurringly

mentioned, the errors were chosen based on their frequency and linguistic relevancies attested in existing L2 literature. Table 6 provides the overall summary of the segmental errors found through the supervised recognition.

Fricative substitution errors: Fricatives are reported to be difficult for Korean L2 speakers as the language lacks rich fricative inventories in English (Hong et al. 2014). Voiced fricatives are particularly challenging, with the added dimension of voicing distinction absent in Korean. Concerning these two aspects, two substitution routes are documented. Substitution of voicing identity in *Z to S* and substitution of manner of articulation in *DH to D*, *V to B*, and *F to P*. Relevant to the former, Korean alveolar fricatives are all voiceless ([s] and [s^h]). The lack of voicing contrast awareness hereafter causes difficulties in articulating [z]. The latter cases commonly involve a shift towards homorganic plosives. These errors are fostered by a comparatively richer inventory of plosives in Korean. The acoustic distance between the plosive counterparts and existing phonemes in the speakers' mother tongue is closer, which leads learners to approximate foreign fricative sounds to an acoustically more familiar plosiveness (of existing inventories in L1; [t],[p],[p^h]).

Liquid substitution errors: Alternations between L and R phonemes can be viewed under the same lighting. Unlike English exhibiting two unique lateral /l/ and rhotic /ɹ/ liquids phonemically, Korean liquid inventory consists of only one element. Rhotic (tap) [ɾ] and lateral [l] come as an allophonic variation of this single phoneme /ㄹ/. This could lead to confusion in producing English liquids since the two modes of articulation do not always show complementary distribution as in the learner's L1. Under this context, the articulation accuracy of liquids is deeply correlated

with the Korean L2 speaker's pronunciation level (Kim et al. 2019), making them a relevant target for sub-segmental examination.

Vowel substitution errors: The selected vowel substitution errors can be summarized under three different reasonings. The first type occurs due to the absence of tense-lax distinction in the Korean vowel system (Yang et al. 2013). This can account for the substitution of IY to IH, IH to IY, AE to EH, and EH to AE. The lack of distinction relates two phonemes bidirectionally; hence, every two substitutional directions are to be evaluated as a pair. The second vowel substitution results from a lack of corresponding inventory to the English mid-back vowel AH (Ku & Oh 2001). When mapping this foreign sound to familiar native phonemes, AH bears between-categorical traits encompassing the boundaries of /ㅏ/and /ㅓ/. /ㅏ/ also forms a close acoustic distance with AA, which was detected as the most dominant substitution target with 4452 occurrences. The third instances are subsumed under the diphthongal production category (Choi & Oh 2021). English diphthongs OW and EY are both non-present in Korean, often leading L2 Korean speakers to reduce them to monthlong AO and EH that each bear closer value with present phonemes /ㅓ/and /ㅔ/.

Phonetic Category	Substitution pattern	Phonetic Category	Substitution pattern
Fricatives	Z to S	Vowels	IH to IY
			IY to IH
Fricatives	DH to D	Vowels	AE to EH
			EH to AE
Fricatives	V to B	Vowels	AH to AA
Fricatives	F to P	Vowels	OW to AO
Liquids	L to R	Vowels	EY to EH
	R to L		

Table 6 selected list of substitution errors

4.2.2 Discovered Sub-Segmental Patterns

For every segmental substitution secured in Table 6, the sub-segmental pattern analysis is sequentially conducted through internal and external inspections. The initial internal inspection concerns discovering the three most dominant index pairs, followed by external inspection to judge their attributes and uniqueness using the L1 reference data. Accordingly, the analysis followed the order of pinning down the identified representative index and documenting two-way comparisons. This analysis schema is graphically reiterated in Figure 28 to aid understanding.

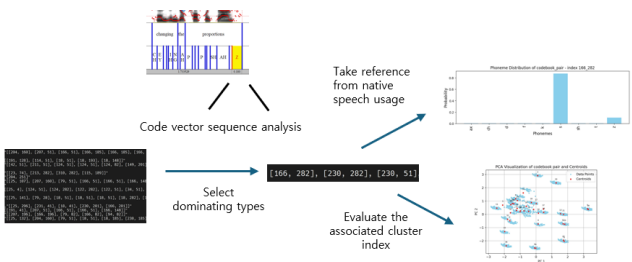


Figure 28 sub-segmental pattern analysis schema

To illustrate an example of its application, the analysis process for the Z to S substitution is provided. Among sound samples of Z to S error, the 3 most dominant indices were [166, 82], [18,51], and [230, 51].

Dominant index	[166, 82]	[18,51]	[230, 51]
Associated cluster	21	Not present in native speech	28

Table 7 internal pattern discovery result of Z to S substitution

The associated cluster ID of the pair [166, 81] is 21, whereas [230, 51] belongs to 28. [18, 51] did not have a native speech occurrence, so it was impossible to measure similarity with the other two indices. Cluster 21 and 28 are separated by a considerable amount, as noted in the cluster

visualizations below. Hence, it is safe to assume that the Z to S substitution error accompanies at least two sub-segmental patterns.

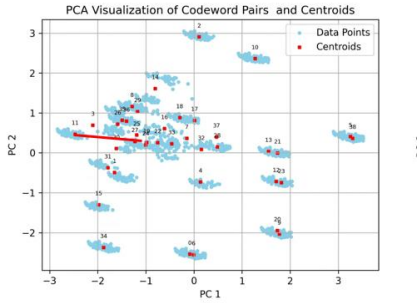


Figure 29 plotting of Z to S substitution dominant index

Next, we check the attributes of each index. Figure 29 shows what phonemes each paired indices corresponds to in L1 TIMIT. Because [18, 51] did not have any native speech example, the paired reference was substituted with examining what each codebook index (codebook1 index 18 and codebook2 index 51) represents.

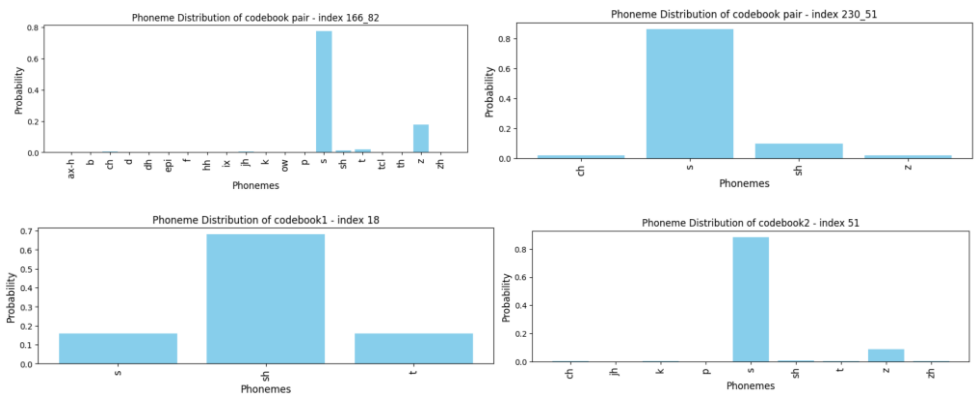


Figure 30 attributes of the discovered index in Z to S substitution

[166,82] and [230, 51] carries a difference in the secondary probability distribution. For [166, 82], the second highest probability is z, whereas it is sh in [230, 51]. [18,51] assumes a middle ground between these two. 18th codeword in codebook1 allocates the highest probability to sh, while index 51 in codebook2 is a resemblance to [166, 82]. Within such between-index positioning, [18,51] gears slightly more towards [230,51],

concerning the 1) overlap in the second codebook index and 2) the codebook1 index's highest association with sh, which wins a secondary place in the probability distribution of [230,51]. In the end. three patterns are identified that form gradience in the voicing spectrum. The pertinent pattern-differentiating traits came from the comparative association rate with voiceless sh and voiced z.

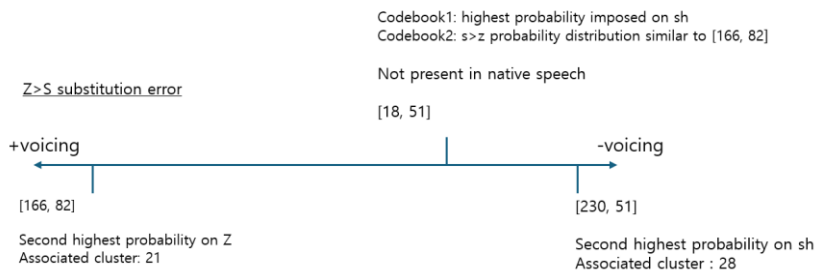


Figure 31 discovered patterns in Z to S substitution

Applying the same analysis strategy to the rest of the 12 segmental substitutions arrives at the following three overarching findings. These were the common threads underlying subsegmental pattern typologies.

A Subject of Gradience

The first finding was that the relationship among sub-segmental patterns could be traced along the assumed degree of the changed articulatory trait. We have already seen that in the Z to S example, voicing identity in charge of the shift also formed the axes of the substitution spectrum. Interpretative evidence came from the second-highest probability in native phoneme distribution. Among present patterns, the associated phonemes moved from sh [-voicing] to z [+voicing]. Along with the substitution of voicing identity, 5 other substitution routes form our 13

lists of errors, as each trajectory entails relevant pattern-differentiating traits.

Substitution of the manner of articulation: In 4.2.1, the second substitution trajectory of fricatives concerned a switch in the manner of articulation. Namely, fricatives are replaced with homorganic plosives in DH to D, V to B, and F to P. The relevant articulatory trait in this shift is continuity, which formed the error pattern continuum. Across 3 examples, there were recurring dynamics among constitutive dominant patterns. Native phoneme distribution of patterns moved from a relatively higher association with plosives and silence markers [-continuity] to a higher association with fricatives and vowels[+continuity]. An example of the former [-continuity] end in DH to D and V to B is provided in Figure 32, whereas the [+continuity] end counterpart is provided in Figure 33. For the reference of where each pattern lies in the spectrum, the full scale is outlined in Figure 32.

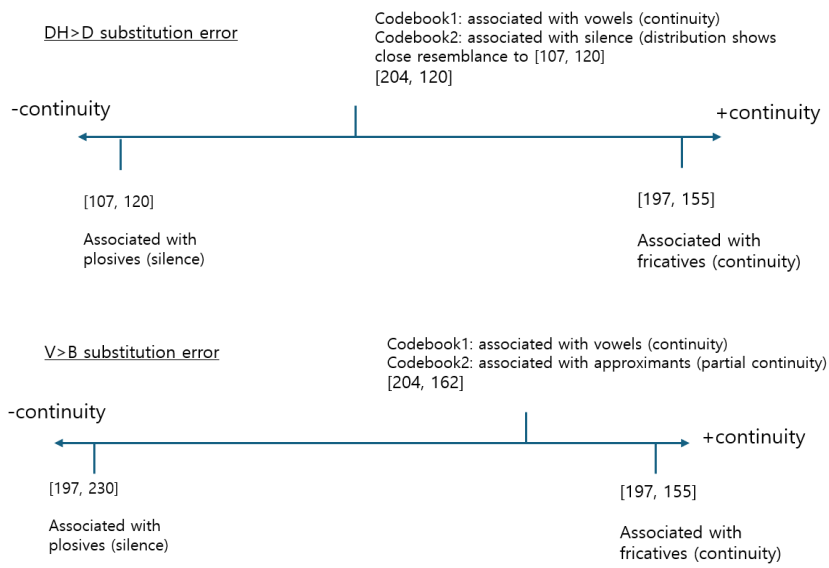


Figure 32 discovered typologies in DH to D and V to B substitutions

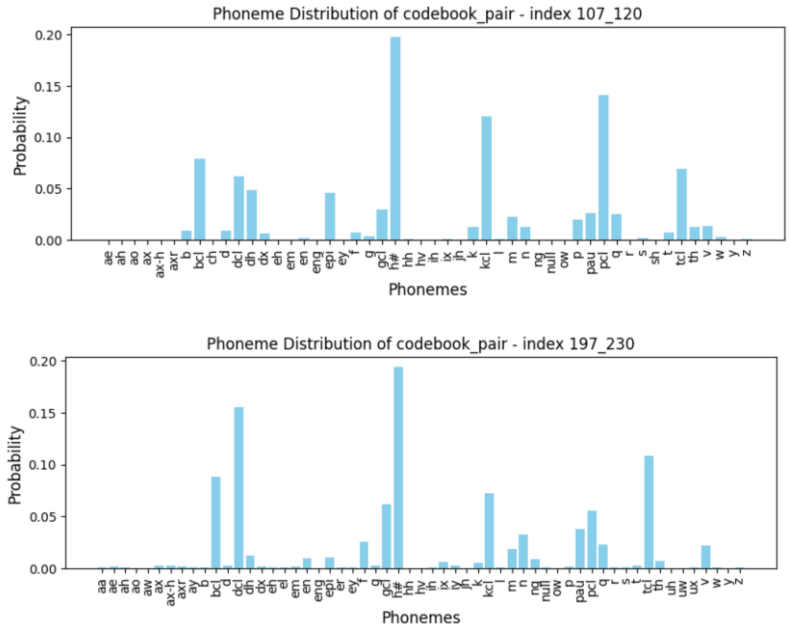


Figure 33[−continuity] end in the substitution of the manner of articulation

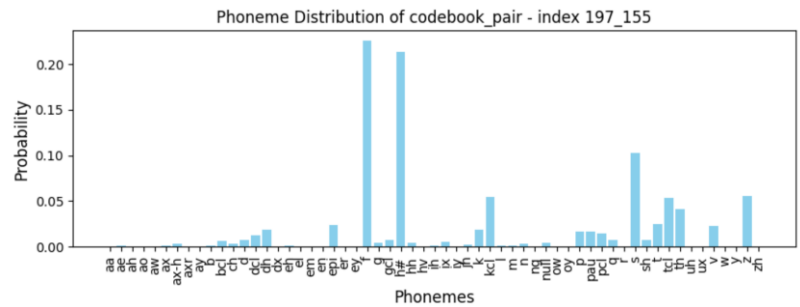


Figure 34[+continuity] end in the substitution of the manner of articulation

Substitution of laterality: The switch of lateral to rhotic liquid concerns a change in the channel of airflow. Lateral liquid guides the airflow towards the side of the tongue by approaching the tongue tip near an alveolar ridge. This process of creating a lateral channel concerns the trait laterality and is absent in English alveolar approximant /ɹ/. The bunched r rather retracts the tongue tip, preventing even the slightest chance of obstruction. It involves a lesser degree of constriction, making them more resonant and acoustically closer to vowels. Thus, when more probability is shared with

vowels, it would indicate a comparatively [−laterality] attribute. With this understanding, the native phoneme distribution moves from having more lateral [+laterality] to rhotic and vowel [−laterality] association (or vice versa as the substitution is bidirectional). Examples of [−laterality] end, [+laterality] end, and the full pattern spectrum are presented below.

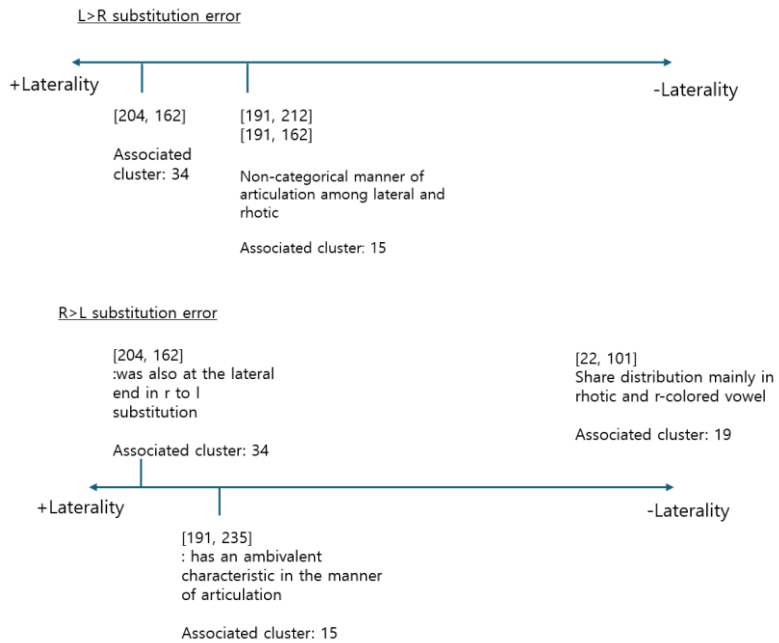


Figure 35 discovered typologies in liquid substitutions

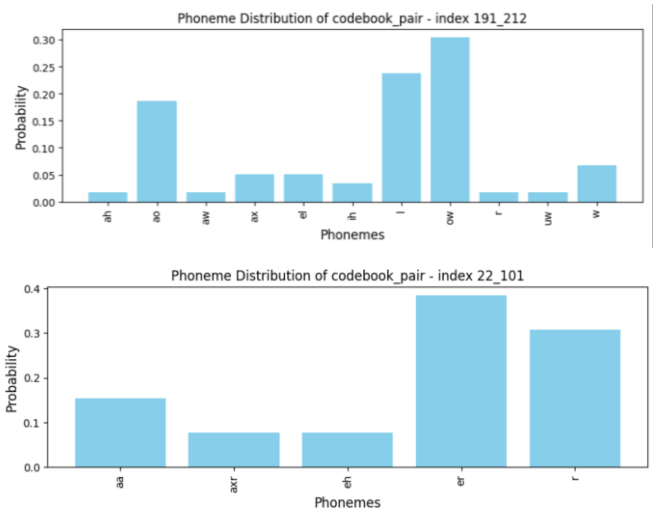


Figure 36[−laterality] end in the substitution of laterality

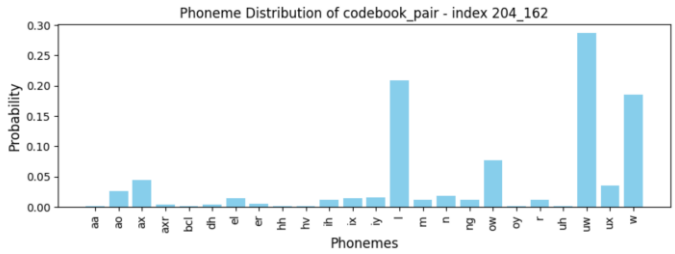


Figure 37[+laterality] end in the substitution of laterality

In L to R, there is a cluster overlap between two dominant patterns [191, 212] and [191, 162]. Both belonged to cluster 15 and were thus merged into a single pattern. [191, 162] was left out of the description as it had only 3 recorded instances in L1. Such lack of representation is due to the non-categorical nature of sound, as will be further explained in section B.

Substitution of vowel height: In dominant patterns of AH to AA, singularity is marked by the relative association rate with AO and AA in native speech. The two phonemes primarily differ in vowel height with AO sharing the same height as AH. Hence, the gradience is formed along the vowel height spectrum, being in line with the changed articulatory trait itself. In Figure 37, patterns gradually move from bearing relatively [-high] to [+high] characteristic. The gradient nature is emphasized by the shared probability with AO in the pattern of [-high] end (Figure 38).

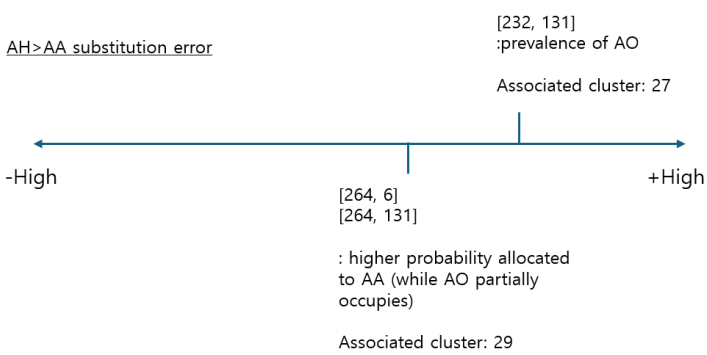


Figure 38 discovered patterns in the mid-back vowel substitution

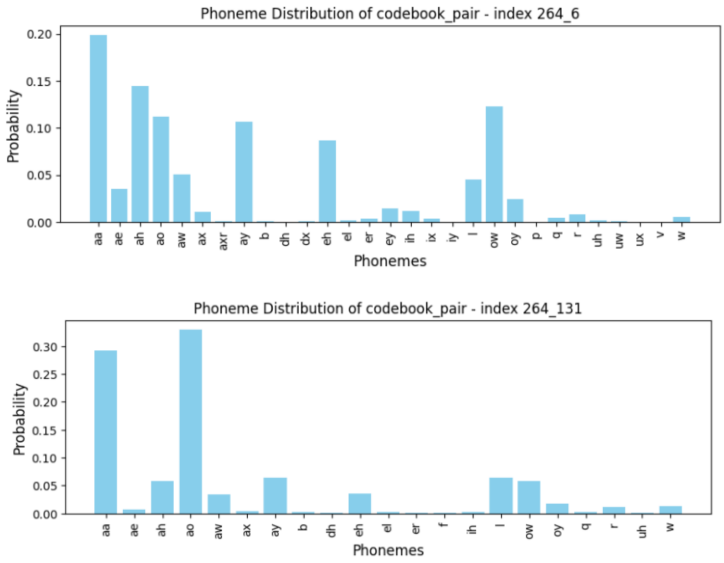


Figure 39[−high] end in the substitution of vowel height

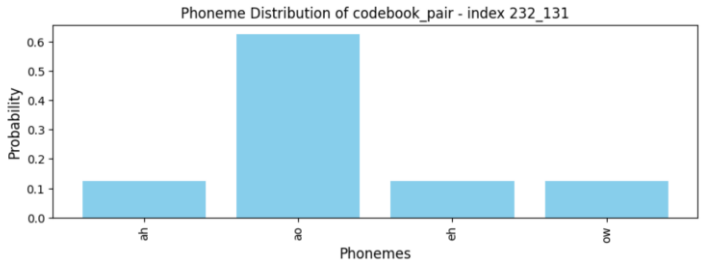


Figure 40[+high] end in the substitution of vowel height

The mid-vowel height relocation also experiences a pattern overlap. [264, 6] and [264, 131] both belong to cluster 29 which shares a higher probability with AA than [232, 131]. Their distance with the [+high] end pattern is not large, concerning the proximity of cluster 27 to 29.

Substitution of tenseness: Gauging the relationship among subsegmental patterns of IH to IY, IY to IH, and AE to EH was difficult with the standard phoneme distribution−based comparison. The reason was that the observed dominant indexes had little to no native speech presence. Alternatively, pattern attributes were identified by analyzing individual

indices, which spelled out a cohesive trend among codebook combinations. While codebook1 displayed an identity of laxness, codebook2 showed an identity of tenseness. This was confirmed by calculating the tense-to-lax ratio of associated native phonemes. Figure 41 shows the used tense and lax vowels for calculation along with computed ratios of pertinent indices.

Tense	ae, aw, ay, ey, iy, ow, oy, uw
Lax	aa, ah, ao, ax, axr, eh, er, ih, ix, uh, ux

Codebook1 (displays laxness)
Tense-Lax ratio: 191 (0.511) > 42 (0.375) > 22 (0.315)
Codebook2 (displays tenseness)
Tense-Lax ratio 234 (9.744) > 268 (2.981)

Figure 41 tense-to-lax ratio calculation

The rarity is likely attributed to these conflicting identities. The following are the observed combinations and their amount of presence in L1 reference data. [191, 234] had four occurrences (iy:2. ey:2), [22, 268] had two (eh:1, ey:1), [42, 234] and [22, 234] had none. Referring to the tenseness ranking, present patterns can be scaled as Figure 42.

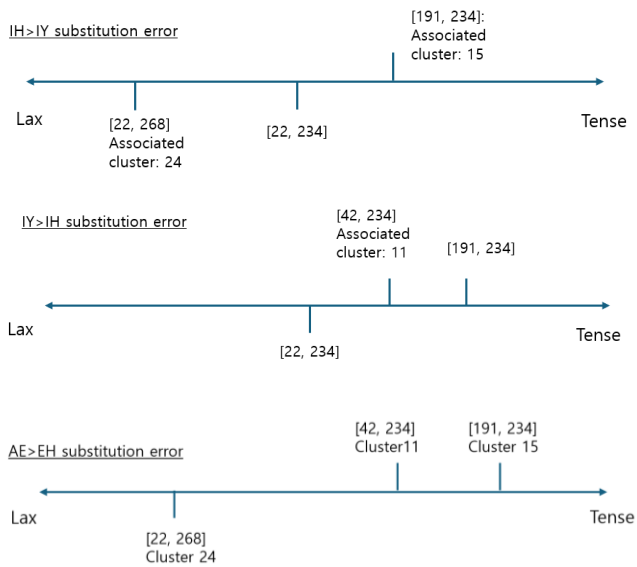


Figure 42 discovered typologies via tenseness calculation

The scaling here aggregates the rankings in two codebooks. [22, 268], for instance, occupies the lowest end in IH to IY with the multiplied value of 0.919 (0.315*2.981). On the other hand, [191, 234] occupies the highest end with 4.979 (0.511*9.744). These tenseness rankings are consistent with the Euclidean distance between code vectors. [22, 268] and [191, 234] each belong to cluster 15 and 24 that are considerably apart. Further, the distance between [22, 268] and [191, 234] is indeed larger than the other two pairs, while codebook1 ranking is in line with the code word interval shown in the rightmost visualization graph of Figure 43.

- [22,268]-[22,234] 5.762
- [22, 345] – [191, 234] 4.715
- [22, 268] – [191, 234] 7.44

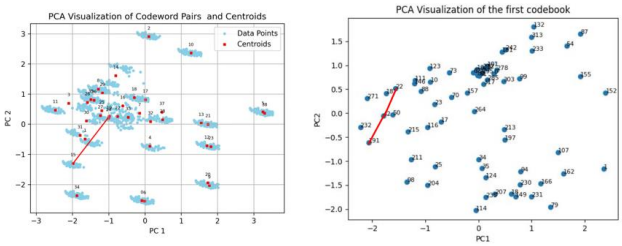


Figure 43 consistency of tenseness ranking with Euclidean distance

Meanwhile, EH to AE came as an exception, as their patterns did not adopt any of the calculated indices. They also lacked distinction as three index pairs were either associated with cluster 29 or 8, which were extremely close to one another. Ultimately, they came under a single unified pattern, a process illustrated in the first two charts below.

Dominant indexes	[264, 153]	[264, 64]	[264, 315]
Associated cluster	29	29	8

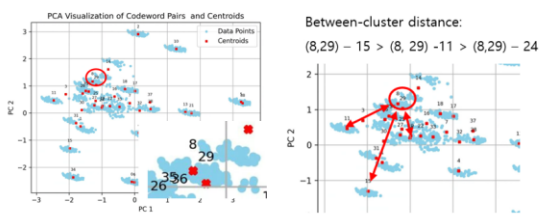


Figure 44 EH to AE substitution

The merged pattern leans more towards laxness than tenseness. The rightmost graph of Figure 44 delineates the distance between the merged location and three clusters scaled on the tense–lax spectrum in AE to EH (Figure 42). Accordingly, the distance is the shortest with the most lax cluster 24, while the longest with the most tense cluster 15. The intermediary cluster 11 takes a middling position in length as well. The uniqueness of EH to AE will be further discussed in Chapter 5.

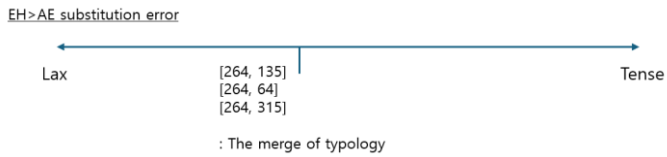


Figure 45 discovered pattern in EH to AE substitution

Substitution of production focus: As diphthongs inherently involve a shift in pronunciation position, characteristics derived from two positions potentially lead to conflicting identities. Canonical pronunciation displays resolved conflict towards the nucleus position, whereas the dispersed focus could lead to monophthong reduction error. Figure 46 charts the comparative position of the supposed articulation trajectory in relation to the substituted place of articulation.

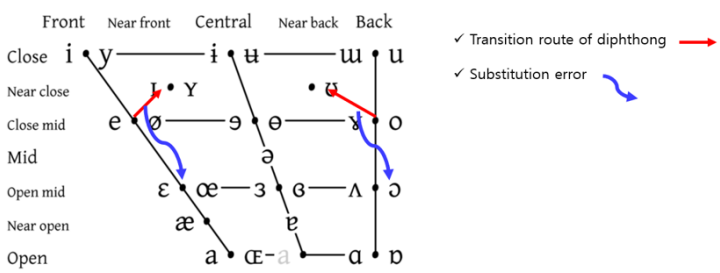


Figure 46

Reflecting this nature of positional conflict, patterns are scaled on the spectrum of the degree of resolved backness ambiguity. As the desired focal point of production is the diphthong nucleus, OW and EY are respectively judged on the standard of posterior and anterior resolution. This mirrors the relative position of the diphthong within the movement. Similar to tenseness substitution, attributes of a conflict-mediating nature are identified by calculating the identities of the individual code word. Figure 47 displays the used phonemes for calculation alongside computed ratios of posterior (back-to-front) and anterior (front-to-back) resolution. Instead of limiting the positional identification to vowels, an entire catalog of sounds using the front and back cavities was considered.

back-to-front ratio

Codebook1 191 (1.687) > 232 (0.7) > 22 (0.559)

Codebook2 212 (2.084)

Back	aa, ae, ah, ao, aw, ow, uh, uw, ay, oy, w, ux, r, hh, q, k, hv, ng, g, gcl
Front	eh, er, ey, ih, iy, y, l, ix, el, dx, b, bcl, f, p, n, m, v, dcl, dh, pcl, tcl, t, z, nx, em, th, s

front-to-back ratio

Codebook1 234 (11.363) > 319 (3.606)

Codebook2 42 (2.666) > 22 (1.788)

Figure 47 positional identity calculation

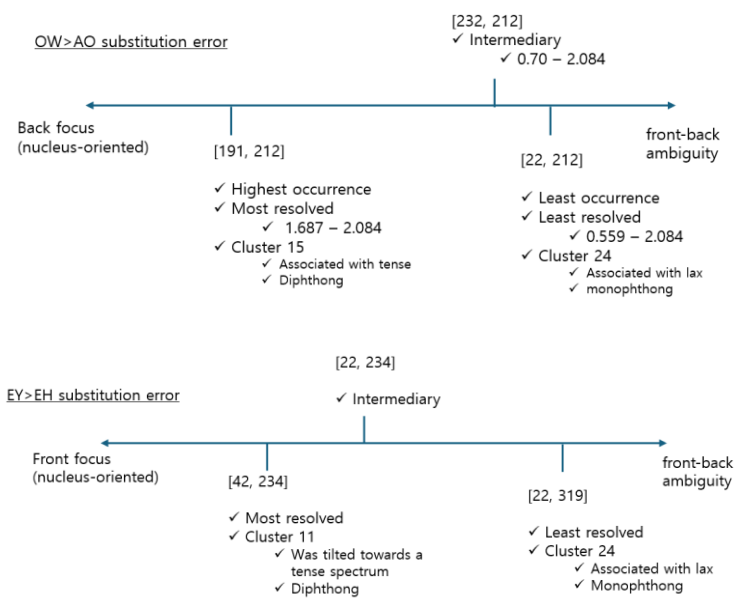


Figure 48 discovered typologies in diphthong reduction errors

Ranking in the spectrum is likewise grounded in the aggregation of computed values of the paired code words. In the OW to AO spectrum, the pairing numbers are annotated, ranging from the highest (1.687−2.084) in [191, 212] to the lowest (0.559−2.084) [22, 212]. These calculations can be verified from the perspective of the rarity of occurrence and tenseness identity. Regarding the former, the index frequencies in L1 data aligns with between−pattern distances. In Figure 48, the most compatible pair [191, 212] has multiple presences, while only two occurrences (ah:1, ow: 1) are observed in [22, 212], when index 22 with the last backness distribution engages. The associated cluster of less frequent pair [232, 212] and [22, 212] are close to each other compared to [191, 212].

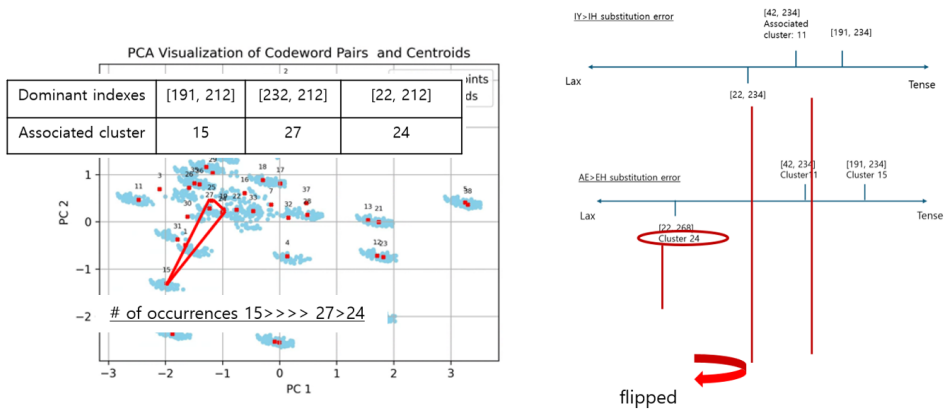


Figure 49 verification of the production focus scaling

Moreover, there is a correlation between diphthongal property and tenseness. The cluster previously defined to constitute the tense end of the spectrum (15) is associated with the most resolved pair exhibiting the closest value to a diphthong. Conversely, the cluster positioned at the lax end of the spectrum (24) is associated with the least resolved pair bearing the closest value to monophthong. One can visually spot this trend in the

ranking of EY to EH, which is essentially a flipped ordering of tenseness substitution ranking.

B Intermediary Typology

Another foundational characteristic prevalent across sub-segmental patterns was that the intermediary typology was non-categorical. This incoherent positioning was rendered by assuming ambivalent identities in two codebooks. As seen from the Z to S example, if one displayed a + value of the changed articulatory trait (i.e.[+voicing] in codebook2), the other displayed an opposite negative value (i.e. [-voicing] in codebook 1) by comparison. The contradictory pairing was the cause of rare occurrences in native speech, attesting to the L2 particular nature of non-categoricity. Intermediary of Z to S, [18, 51], had no presence. Below are other examples organized by their substitution routes.

Substitution of the manner of articulation (Fricative to Plosive): Referring to Figure 32, intermediaries of DH to D and V to B were [204, 120] and [204, 162]. Reflecting the recurring dynamic, they both belonged to cluster 34 while displaying comparatively [+continuity] identity in codebook1 and [-continuity] identity in codebook2. Figure 50 shows that the 204^{th} codeword of the first codebook has a primary association with vowels involving continuous airflow. Conversely, the 120^{th} codeword of the second codebook mainly shares distribution with silence accompanied by airflow obstruction. Although codebook2 index 162 is affiliated with approximants, they still bear comparatively less continuous identity than vowels of codebook1. Since [204, 162] resolves the discrepancies between vowels and approximants, the ambivalence is geared more towards the fricative end compared to DH to D. Therefore, the gulf between values was

not as large as the V to B intermediary showed more presence. This reflects the relationship between frequency and level of contradiction.

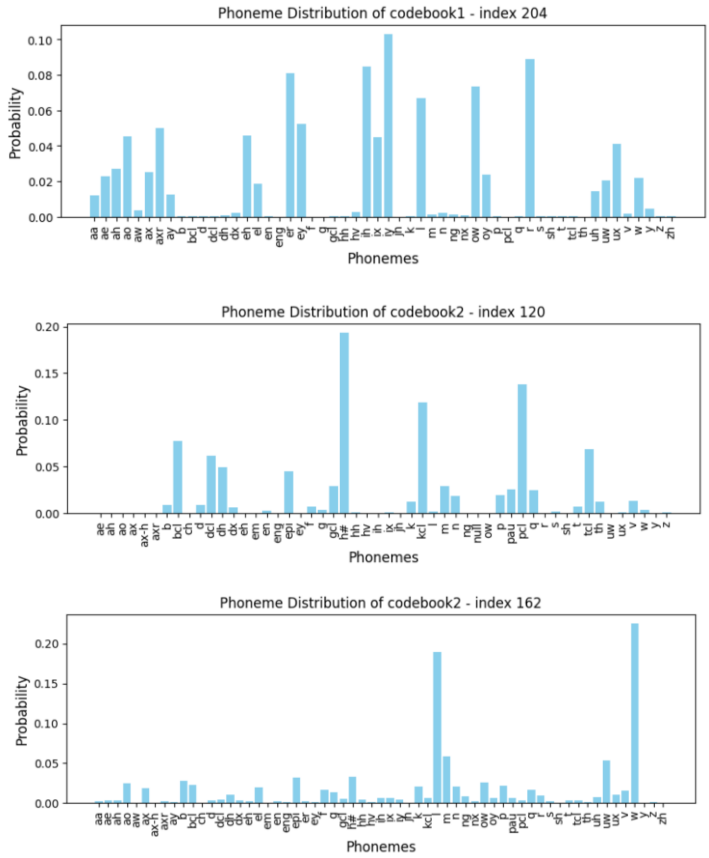


Figure 50 intermediaries of voiced fricative to plosive vying

Substitution of laterality: In both directions of liquid substitution (Figure 35), intermediaries showed rare occurrences in native speech. [191, 162] of L to R appeared twice, whereas [191, 235] of R to L appeared three times. The rarity also concurred with the unlikely pairing of conflicting attributes. Codebook1 bore [−laterality] attributes with higher rhotic and vowel association, whereas codebook2 had greater lateral identity.

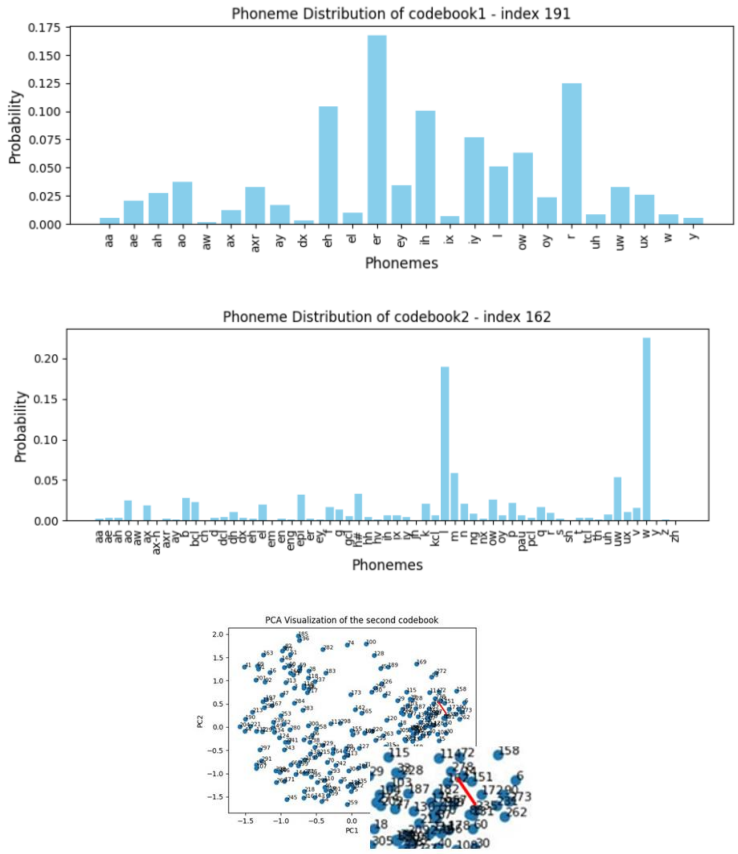


Figure 51 intermediaries of liquid substitution

Figure 51 explains how the intermediary pattern of liquid is the literal instantiation of non-categorical sound between rhotic and lateral. The two codebook2 indices 162 and 235 were, in fact, adjacent to each other as can be noted from the plotting in raw vector visualization.

Substitution of vowel height: Lastly, it was mentioned that the AH to AA intermediary displays a gradient move from the [+high] end (Figure 38). The prevalence of AO in [232, 131] (Figure 40) permeates through the intermediary [264, 131] (Figure 39), which albeit with AA prominence, has a partial AO association. The coexisting articulatory traits were, in fact, encoded in the individual codebook distribution shown in Figure 52. Namely, codebook1 and 2 each assumes [-high] and [+high] identity.

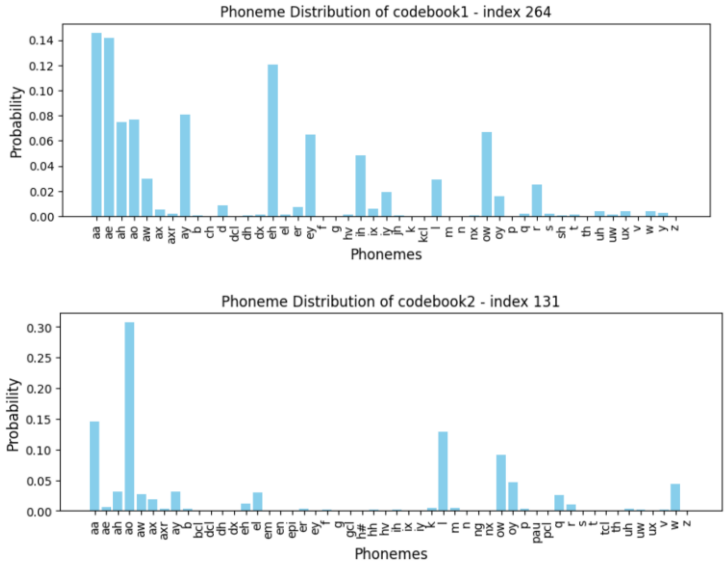


Figure 52 intermediary of vowel height substitution

Lastly, intermediaries of tense-lax substitution and diphthong reduction errors are of a different nature, as the gradience itself was based on the degree of conflict. Tenseness and positional resolution were ranked in proportion to the homogeneity of the two codebook values. Therefore, the lowest end of the numerical scale, rather than the intermediary, concurred with the highest level of contradiction and least occurrence. Nevertheless, the correlation between unlikely combination and rare occurrence still holds, as we shall see in the overall review of front vowel substitutions in Chapter 5.

C Distributional Asymmetry

The third core finding is that typological distributions are skewed towards the most approximate sound available in the learners' mother tongue. The finding can be viewed at two levels: 1)distribution rate across different substitution types and 2) within-substitution pattern distribution.

Regarding the former, one can first refer to the fricative to plosive

substitutions. We have previously omitted an illustration of F to P to emphasize a point here. Figure 53 gives the visual plotting of associated cluster indices for each of the three errors. Compared to the former two voiced fricative varieties, F to P involves smaller between-pattern distances along with an overall rightward shift.

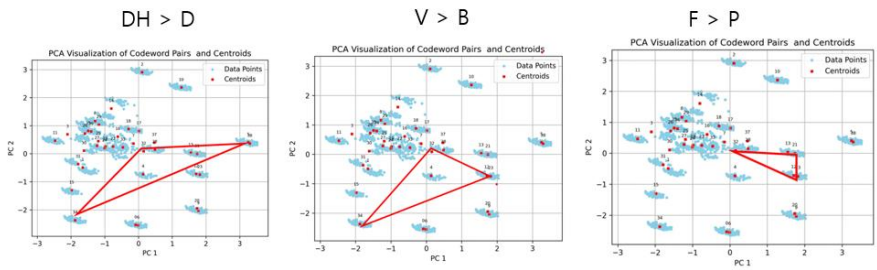


Figure 53 fricative to plosive dominant pattern dynamics

These two aspects are reflected in the spectrum of Figure 54. The plosive to fricative vying has scaled down with a more homogenous distribution towards fricatives. [197, 155] setting the [+continuity] fricative ax in DH to D and V to B (Figure 31) now positions itself as the [-continuity] lowest end. The rest two indices, [166, 196] and [197, 284], allocate their probabilities almost entirely to fricatives. Figure 55 shows that t and #h in [197, 284] is the only room for exception here.

Figure 54 discovered patterns in F to P substitution

The reason follows that Korean L2 learners, with voiceless fricatives

already present in their sound system, should face fewer difficulties with pronouncing fricatives of the familiar phonation type. This logic sounds fitting concerning how representative indexes share a considerable portion with alveolar fricative [s] that is the closest to the phoneme /ㅅ/ in Korean.

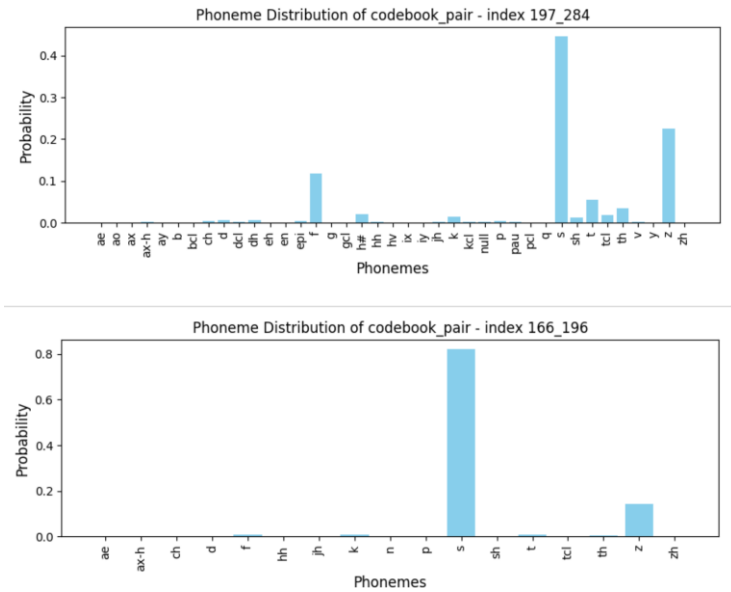


Figure 55 scaled down gradience in F to P

In the end, F to P being markedly skewed towards [+continuity] than DH to D and V to B, is related to the closest counterpart in L1. The mapped corresponding phoneme is voiceless regardless of the articulation target. Therefore, voiceless F displaying the closest distance with the present inventory induces less confusion and, in turn, incurs less dispersion.

The dispersion rate in bidirectional substitution dynamics likewise reflects the interaction with the native sound system. In IH to IY and IY to IH in Figure 42, lax to tense error incurs more dispersion while the tense to lax typologies are more aligned with the canonical sound. In AE to EH and EH to AE in Figure 42 and 44, tense to lax error incurs more dispersion while lax to tense typologies are more aligned with the canonical sound.

This is related to the comparative foreignness of the target among the paired substitution directions. Tense IY and lax EH are closer to the existing Korean phoneme inventory /ㅣ/ and /ㅐ/ than their lax and tense counterparts IH and AE (Ku & Oh 2001). Ultimately, more foreign target incurs greater dispersion.

The skewed distribution within substitution can be found in the case of voicing identity substitution (Figure 30), liquid substitution (Figure 35), and mid-back vowel height shift (Figure 38). To begin with, our first introduced example Z to S had the intermediary skewing towards the [-voiceless] end. This relates to the corresponding native phoneme /ㅅ/ being voiceless. In liquid, distributions are skewed towards the laterality. In L to R, cluster overlap gears towards the lateral end. In R to L, the intermediary pattern bears a closer [+laterality] identity than the [-laterality]. The asymmetry reflects greater difficulties involved with producing the unobserved variety of rhotic [ɹ] in the learner's mother tongue. In AH to AA, distributions are skewed towards the high end, which is related to the most adjacent sound in native speech, /ㅋ/, being closer to AH than AA.

Chapter 5. Discussion

Our analysis result first confirmed that code vector features used by L2 speakers differ from the set employed in native speech. While greater usage overlap coincides with increasing proficiency, one can expect that the more pronunciation deviation there is, the higher the chance for the speech to utilize unused varieties in L1. Such was the case for typological overlap in front vowel substitutions. In four substitution errors in the left chart of Table 8, every pattern constituting one substitution has at least one co-occurrence in another error type. The used varieties were also commonly rare in native speech. [22, 234] was unobserved, while [42, 234] and [22, 268] appeared once and twice each. [191, 234] records 4 instances. Given the overlap and rarity, L2 front vowel substitutions seem to collapse into a mode disparate from L1 renditions. The finding concurs with the vowel space analysis in (Ku & Oh 2001), whereby Korean English vowel space is reported to be smaller, particularly along the frontal region. Accordingly, Korean speakers are not versed in manipulating oral constriction required for the accurate articulation of English front vowels. Tongue tip movement is more limited, creating acoustic distance between IH, IY, EH of learners and that of American native speakers.

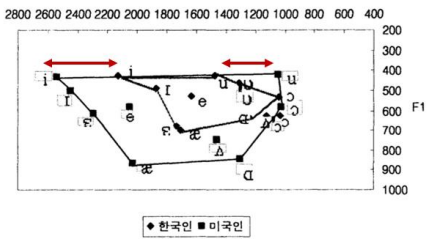


Figure 56 vowel space analysis in Ku & Oh (2001)

This acoustic trend is reflected in the code vector distributions across L2 analysis and L1 reference corpora. We have calculated the Euclidean distance between the most dominant index of front vowel

segments in L2 erroneous speech and L1 TIMIT data. In Table 8, selected indexes for calculation are emboldened. Pairwise distances for IH, IY, and EH were 8.14, 7,80, and 7. 476/7.79 each. EH has two recorded distances as substitution AE→EH and EY→EH were associated with a different dominant index.

For comparison, vector-wise L1-L2 Eucledian distance among back vowels involved in substitutions were also measured. Dominant indexes used for calculation are likewise, emboldened. Overall, distances between the representative index of the L1 phoneme and the corresponding erroneous segment in L2 are a lot smaller in back vowel substitutions than in front vowel pairs. Typological distances in AA and AO were 5.626 and 5.264 each. Such difference reflects the greater deviation occurring in front vowel space. Note that back vowel variants were also more frequently observed in native speech, which proves that a smaller acoustic distance concurs with better correspondence. They further do not experience typological overlap, as the lack of distinction is primarily caused by a limited range of movement in the frontal cavity.

IH> IY	[22,234], **[191, 234],** [22, 268]
IY>IH	**[191,234],** [22,234], [42,234]
AE>EH	**[22,268],** [42,234], [191,234]
EY>EH	[42, 234], [22, 234], **[22, 319]**

AH > AA	**[264, 6],** [264, 131], [232, 131]
OW > AO	[22, 212], [232, 212], **[191, 212]**

Table 8 dominant index pairs of vowel substitutions

Another peculiar finding in front vowel substitution is that EH → AE is immune to the overlap while experiencing cluster merger. Substitution leading up to AE forms a unique identity lacking sub-segmental variation. The situation is comparable to the findings of Yang et al. (2013), where a comparative distance between Korean English and

American English front vowels was measured. Referencing the vowel plot in Figure 55, one can notice that the AE variation is greater in American English in blue notation, while for EH the reverse holds.

This difference in the comparative dynamic in EH and AE relates to the present code vector observation. On one hand, greater L1 variability of AE creates room for L2 error pattern correspondence. Hence, the used varieties in EH to AE had more native speech presence compared to other front vowel substitution errors. On the other hand, because the receptive field of L2 is comparatively narrower, differences within AE could be overlooked when interpreted in an L1 standard i.e. with L1 reference material. This would be the cause of the cluster merger. Meanwhile, as the situation is the opposite for EH, AE to EH error has a lower native speech correspondence while being more dispersed.

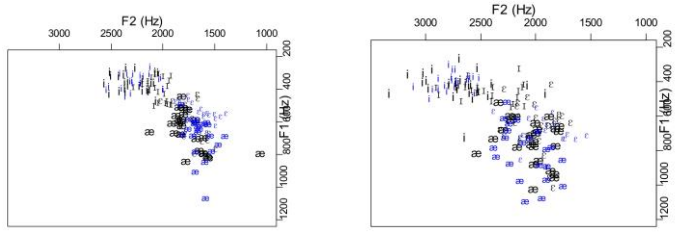


그림 2. 23명의 한국인과 미국인 남성이 발음한 전설모음의 F1과 F2로 나타낸 분포
Figure 2. A vowel chart of F1 by F2 of the front vowels produced by 23 Korean and American speakers

그림 3. 24명의 한국인과 미국인 여성이 발음한 전설모음의 F1과 F2로 나타낸 분포
Figure 3. A vowel chart of F1 by F2 of the front vowels produced by 24 Korean and American female speakers

Figure 57 vowel plot in Yang et al. (2013)

On top of these peculiar findings, the general observations in 4.2,2 imply that the discovered patterns are linguistically interpretable. The between-categorical positioning in the error continuum could be calculated through the value of the substituted articulation trait, whose measurement expounded how non-categoricities were rendered. In this vein, the uncovered sub-segmental patterns reflected the acoustic distance in L2 variations established in the existing literature. Unfortunately, direct

comparisons with prior works on non-categorical error pattern discovery (Wang & Lee. 2013, 2015; Li et al. 2018; Li et al. 2020; Mao et al. 2018) were not viable as they did not concern the L1-L2 pair of the current interest. Nevertheless, concerning all the observations we made, this work affirms the adequacy of code vector-based non-categorical pattern analysis.

Lastly, the final step in actualizing the goal of automated sub-segmental feedback leads to a discussion on how we may use the obtained information for finer judgment. Possible paths of suggestion include comparing the codebook sequence of L1 and L2 and assessing the misalignment information. Lee et al. (2012, 2013) have introduced an established methodology to detect mispronunciation by comparison in an unsupervised manner. This work found that just as raw acoustic or DBN posteriors, SSL representation code vectors hold discriminative power against nativeness. We may take advantage of such L2 discernability to automatically spot areas of mismatch. The final decoding format, however, should be more gradual and nuanced. To suffice this, Hu et al. (2023) proposal of restoring continuous representation from the codebook prior could be considered.

Chapter 6. Conclusion

This work recreated the unsupervised L2 error pattern discovery experiments in previous literature using an SSL representation code vector. In this endeavor, it seeks to reinforce the unsupervised nature of the pipeline and conduct sub-segmental analysis with an unprescribed acoustic unit. In implementation, we adopted an unsupervised model for feature extraction, departing from the previous supervised framework. The learned representation, discrete latent of Wav2Vec2.0, was a descriptive mechanism uninhibited by phonemic categorization.

With these goals in mind, we attempted to answer questions on whether the chosen acoustic unit is suitable for sub-segmental analysis, and if so, how it would capture subtle variations within each segmentally defined error type. From the first experiment, one could spot the difference in used inventory, and how the encoding units of L2 differed from L1. The second experiment revealed that this difference resulted from an unlikely codebook combination in L2 assuming conflicting characteristics. The pertinent two opposing identities each formed the opposite ends of the typological spectrum, while the unlikely combination was the prime example of non-categorical sound. Consequently, a higher degree of phonetic divergence coincided with the increased usage of such L2 particular non-categorical index. Moreover, distributions of sub-segmental typologies reflected the acoustic proximity of corresponding L1 phonemes to two L2 segments participating in substitutions. Thus, the way code vectors encode L2 variation is phonetically relevant.

In the end, the Wav2Vec2.0 code vector is a valid tool to uncover sub-segmental gradience in grossly categorized substitution errors. Its attribute bears quantifiable phonetic relevance that allows us to calculate

the between-categorical details. While discovered gradience urges us to re-evaluate L2 substitution errors from a sub-categorical standpoint, discourse on how we may utilize obtained details for more granular feedback is left for future work.

Reference

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33, 12449-12460.

Boersma, P. (2001). Praat, a system for doing phonetics by computer. *Glot International*, 5(9), 341-345.

Cámbara, G., Luque, J., & Farrús, M. (2022, October). Recycle Your Wav2Vec2 Codebook: A Speech Perceiver for Keyword Spotting. In Proceedings of the 29th International Conference on Computational Linguistics (pp. 7166-7170).

Chan, C. A., & Lee, L. S. (2011). Unsupervised Hidden Markov Modeling of Spoken Queries for Spoken Term Detection without Speech Recognition. In *Interspeech* (pp. 2141-2144).

Choi, H., & Oh, M. (2021). Asymmetrical Production of English Diphthongs /eɪ/ and /oʊ/ by Korean Learners of English. Studies in Linguistics,(58), 19-42, 10.17002/sil..58.202101.19

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2020). Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Garofolo, J. S. (1993). Timit acoustic phonetic continuous speech corpus. Linguistic Data Consortium, 1993.

Han, S., Kim, S., & Chung, M. (2024, May). Constructing Korean Learners' L2 Speech Corpus of Seven Languages for Automatic Pronunciation Assessment. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 3772-3781).

Hong, H., Kim, S., & Chung, M. (2014). A corpus-based analysis of English segments produced by Korean learners. Journal of Phonetics, 46, 52-67.

Hu, Y., Chen, C., Zhu, Q., & Chng, E. S. (2023). Wav2code:

Restore clean speech representations via codebook lookup for noise-robust asr. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

Hsu, W. N., Bolte, B., Tsai, Y. H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM transactions on audio, speech, and language processing, 29, 3451-3460.

Kamper, H., Jansen, A., & Goldwater, S. (2016). Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24(4), 669-679.

Kim, R. E., & Rhee, S. C. (2019). A study on English liquids in the rated L2 English speech corpus of Korean learners. Korean Journal of English Language and Linguistics, 19(1), 53-75.

Kominek, J., & Black, A. W. (2004). The CMU Arctic speech databases. In Fifth ISCA workshop on speech synthesis.

Koo, H-S., & Oh, Y-J. (2001). An Analysis of English Vowels of Korean Learners of English and English Native Speakers. Korean Education Inquiry, 16, 1-12.

Lee, A., & Glass, J. (2012, December). A comparison-based approach to mispronunciation detection. In 2012 IEEE Spoken Language Technology Workshop (SLT) (pp. 382-387). IEEE.

Lee, A., Zhang, Y., & Glass, J. (2013, May). Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 8227-8231). IEEE.

Lee, A., & Glass, J. (2015). Mispronunciation detection without nonnative training data. In Sixteenth Annual Conference of the International Speech Communication Association.

Lee, A., Chen, N. F., & Glass, J. (2016, March). Personalized

mispronunciation detection and diagnosis based on unsupervised error pattern discovery. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6145-6149). IEEE.

Li, X., Mao, S., Wu, X., Li, K., Liu, X., & Meng, H. (2018, September). Unsupervised Discovery of Non-native Phonetic Patterns in L2 English Speech for Mispronunciation Detection and Diagnosis. In INTERSPEECH (pp. 2554-2558).

Li, X., Wu, X., Liu, X., & Meng, H. (2020). Deep segmental phonetic posterior-grams based discovery of non-categories in L2 English speech. arXiv preprint arXiv:2002.00205.

Liu, S., Mallol-Ragolta, A., Parada-Cabaleiro, E., Qian, K., Jing, X., Kathan, A., ... & Schuller, B. W. (2022). Audio self-supervised learning: A survey. Patterns 3, 12 (2022), 100616.

Mao, S., Li, X., Li, K., Wu, Z., Liu, X., & Meng, H. (2018, April). Unsupervised discovery of an extended phoneme set in l2 english speech for mispronunciation detection and diagnosis. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6244-6248). IEEE.

Ng, C. L. C. (2017). Merger of the syllable-initial [n-] and [l-] in Hong Kong Cantonese.

Park, A. S., & Glass, J. R. (2007). Unsupervised pattern discovery in speech. IEEE Transactions on Audio, Speech, and Language Processing, 16(1), 186-197.

Ravi, K. K., & Krothapalli, S. R. (2022). Phoneme segmentation-based unsupervised pattern discovery and clustering of speech signals. Circuits, Systems, and Signal Processing, 41(4), 2088-2117.

Wang, Y. B., & Lee, L. S. (2013, May). Toward unsupervised discovery of pronunciation error patterns using universal phoneme posteriorgram for computer-assisted language learning. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 8232-8236). IEEE.

Wang, Y. B., & Lee, L. S. (2015). 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.

Yang, B. (2013). A comparative study of relative distances among English front vowels produced by Korean and American speakers. *Phonetics and Speech Sciences*, *5*(4), 99-107.

Zhang, Y., & Glass, J. R. (2010, March). Towards multi-speaker unsupervised speech pattern discovery. In *2010 IEEE international conference on acoustics, speech and signal processing* (pp. 4366-4369). IEEE.

Zhao, G., Sonsaat, S., Silpachai, A., Lucic, I., Chukharev-Hudilainen, E., Levis, J.M., & Gutierrez-Osuna, R. (2018). L2-ARCTIC: A Non-native English Speech Corpus. *Interspeech*.

국문 초록

L2 발음은 두 음성체계 간 상호작용 아래 실현되기에 단일 음소 범주보다 복합적인 정체성을 지닌다. 이 같은 비범주적 속성은 음소 보다 세분화된 접근의 평가를 요구한다. 그러나 분절 이하 조사는 상당한 전문인력을 동반하기에 데이터로부터 오류 특징을 스스로 찾는 자동화 연구들이 대두되었다. 다만 기존 연구들은 비지도 방식으로 음소 이상의 변이 패턴을 찾기 위해, 지도학습 자질이자 음소적으로 규제된 음소사후 확률을 사용한다는 모순적 한계가 있다. 이에 본 연구는 비지도 학습만을 요하며, 외부 규제 없이 표현 학습으로 습득된 음성 단위 Wav2Vec2.0 코드벡터를 대체 분석 자질로 도입한다. 동시에 기존 자동화 연구의 주요 프레임워크를 유지함으로써, 코드백터가 분절 오류만으로는 정의될 수 없는 발음 변이 양상들을 설명할 수 있을지 탐색하고자 한다.

자질의 발음 오류 표현 적합성 탐색은 사용 빈도 계산을 통한 L2 식별력 검증과 분절 단위 오류 표본들의 열 분석을 통한 유형(패턴) 분석 두 단계로 진행된다. 같은 발화목록을 지녀 내용적으로 통제된 L1 (CMU ARTIC) 및 L2 (L2 ARTIC) 단일 화자 코퍼스에서 화자 별 코드벡터 사용 목록의 빈도를 벡터로 구축해 군집화하고 비교하였다. 아울러 L1 TIMIT으로 파인튜닝된 모델로 L2 NIA037 내 분절 단위 오류 탐지를 실시해 분석에 사용될 표본들을 선별했다. 강제정렬로 오류 음소에 대응되는 음성 구간을 찾아 원모델로부터 소속 프레임들의 코드벡터열이 추출되면, 대표 인덱스 요약, 빈도 계산, 시퀀스 통합을 거친 내부 분석이 진행되며 우세한 패턴들이 도출된다. 패턴들은 같은 L1 TIMIT을 이용해 구축된 참조자료를 통해 최종 해석된다. 이름하 L1음소-코드백터 공동 발생 확률로 음성학적 특성을 유추했으며, L1 데이터에 현존하는 전체 코드벡터들을 군집화하여, 패턴 간 관계성 및 각 유형의 고유성을 판단했다.

실험 결과, L1 및 L2 화자 사이 빈도벡터들의 군집화 분리를 통해 자질의 L2 식별력을 선차적으로 확인할 수 있었다. 두 화자 집단 간 사용목록의 차이는 특히 낮은 L2 숙련도일수록 감소되는 목록 크기로 확인될 수 있었는데, L1 기준으로 학습된 음성 단위가 충분히 활용되기 미흡한 발화 수준 때문이다.

또한 분절 이하 패턴 양상에서 다음 세 가지 공통된 특징이 기록되었다. 1) 첫째 오류 유형들은 변화된 조음 특성의 반영도를 따라 형성된 연속체를 이루었으며, 2) 각 연속체 속 중도 유형은 본 특성에 대해 두 개의 코드북에서 상반된 값을 구현하는 양면적 성격을 띄었다. 아울러 3)유형 분포는 학습자 L1에 존재하는 가장 근접한 소리를 향해 편향되었으며, 더 생소한 조음 특성을 가진 목표음이 더 큰 분산을 유발하였다. 각 발견은 언어학적 이해를 동반한다. 변화된 특성에 따른 점진적 위상은 발음 변이가 분절적 틀로 정의될 수 없음을 보여주며, 특히 중도 유형의 상충된 조합은 명확한 음소 범주로 분류 불가능한 비범주성의 전형이다. 비범주성이 L2 특성인 만큼 해당 패턴들은 L1 데이터에서 희박한 발생빈도를 가지기도 했다. 아울러 L1 음성체계와 관련된 비대칭성은, 발음 변이가 근본적으로 학습자의 모국어가 목표 학습어에 영향을 미치며 발생한다는 점을 반영한다. 결국 코드벡터는 L2 발음의 연속체적 성격을 수치화 할 수 있는 수단으로써, 발음 오류의 점진성을 평가할 대체 수단임을 주장한다.

,