Automatic phonetic classification of low vowel allophones in Taiwanese Mandarin: Comparisons between models and feature sets

Sijia Zhang

1. Background

Machine learning techniques have been widely applied to the task of Automatic Speech Recognition (ASR). To improve the robustness of ASR performance, one important aspect is to enhance the accuracy of phoneme classification. A phoneme is a smallest unit of sound that distinguishes one word from another in a language. Previous studies on phoneme classification have adopted different acoustics-based features, including Mel-frequency Cepstral Coefficients (MFCC) and Mel-spectrogram using speech corpus data (Almekhlafi et al., 2022; Mukherjee et al., 2018). A variety of machine learning models have been evaluated and compared for phoneme classification in different languages, such as support vector machine (SVM) (Yousafzai et al., 2008 for English) and neural network models (Almekhlafi et al., 2022 for Arabic). Nevertheless, machine-learning based analyses for the classification of allophones are relatively limited. Allophones are phonetically distinct variant of a phoneme; a phoneme can be pronounced differently depending on its phonological environment and its position in a word. Thus, in order to refine the current performance of ASR, it is important to understand how particular phonemes surface as various allophones in different phonological contexts, and how accurately these different allophones can be classified based on various features and classifiers. For instance, Bissell (2021) investigated the allophonic variation within the phonemic classes of vowel /e/ and /u/ in Tol using k-means clustering. Using speech corpus data, acoustic properties (i.e., formant frequency values) at different time points were extracted to analyze the quality of the vowels. Piotrowska et al. (2019) compared k-nearest neighbor (KNN) and artificial neural network (ANN) classifiers for the detection of the dark (velarized) allophonic realization of English alveolar lateral /l/.

This study focuses on examining the allophonic distribution patterns of the low vowel /a/ in pre-nasal coda position in Taiwanese Mandarin. Standard Mandarin is described as having the phoneme /a/ surfaced as front vowel [a] before alveolar nasal [n], and back vowel [ɑ] before velar nasal [ŋ] (Duanmu, 2007). Similar to Standard Mandarin, previous research has also provided acoustic evidence for Taiwanese Mandarin that the low vowel /a/ has significantly different F1 and F2 formants before [n] and [ŋ], suggesting that the backness of the vowel vary depending on the following nasal (Chuang, 2017). The current study aims to build classifiers that predicts the allophonic distribution of vowel /a/ in Taiwanese Mandarin. Different feature extraction techniques, including acoustics formant measurements and HuBERT-generated hidden states (Hsu et al., 2021), were compared for the classification task. Two classifiers, logistic regression model and SVM (support vector machines) were trained and evaluated with different feature sets.

2. Methods

2. 1 Task

The task of this study is to classify the two allophonic forms of the low vowel, [a] and [ɑ] in pre-nasal coda environment. According to Chuang (2017), it is expected that the front vowel [a] occurs before [n], and the back vowel [ɑ] occurs before [ŋ]. Thus, the identity of the following nasal provides information about the gold-standard class of its preceding vowel. Using a label encoder implemented from scikit-learn libraries, the two types of following nasal were transformed into numeric values, where [n] was coded as 0, and [ŋ] as 1. The two classifiers, Logistic Regression and SVM, were implemented using the existing implementations from scikit-learn libraries (Pedregosa et al., 2011). They were first trained on two different feature sets, then fine-tuned based on development sets, and evaluated on the test sets.

Two different types of features were extracted and compared. The first feature set includes the acoustic properties of vowels, namely, formant frequencies F1, F2 and F3, corresponding to the articulatory positions (height, backness and rounding) of vowels. The F1 and F2 has shown to be different for [a] and [ɑ] in Taiwanese Mandarin (Chuang, 2017). The second feature set were extracted through HuBERT model (Hsu et al., 2021), containing high-dimensional vectors representing the hidden units of the speech sounds. The classification accuracy based on both feature sets were compared.

Finally, the study uses Uniform Manifold Approximation and Projection (UMAP) to visualize the structure of the feature vectors the represent the two allophones in pre-nasal position.

2.2 Data

2.2.1 The Corpus and Data Preprocessing

The speech corpus used in this study comes from part of the cross-linguistics corpus constructed by the Origins of Patterns in Speech Lab (OoPS-Lab) at the University of British Columbia. The speech samples consist of both read speech and spontaneous speech collected from 5 Taiwanese Mandarin speakers. All the speech data is forced aligned and segmented at both word level and phone level. The speech files and their corresponding transcriptions (.textgrid) were divided into shorter audio files in Praat (Boersma & Weenink 2016), so that they can be imported into HuBERT for feature extractions. Each audio file included a word, ranging mostly from 1 to 3 syllables and lasting less than 1 second approximately. Only words that contained token /a/ in pre-nasal coda environment were selected. In total, the data includes 1538 /a/ tokens.

All the audio files were further processed using PolyglotDB (McAuliffe et al., 2017), where the timing and acoustic information about the target vowel tokens (/a/ in pre-nasal coda position) were extracted and exported into a csv file. The file includes the time where the vowel token starts and ends, the F1, F2 and F3 of the vowel token measured at its midpoint position, the class of the following nasal ([n] or [ŋ]), the sound token proceeding the target vowel, the word that contains the vowel, the gender of the speaker, the speech rate of the utterance where the word occurs (the number of syllables per second).

2.2.2 Feature extractions

Based on the csv file generated from PolyglotDB, two sets of features were extracted (as shown in final_project_feature_sijia.ipynb). The first feature set contains the F1, F2 and F3 of each low vowel token, along with other information such as the following nasal token, the word where the vowel occurs, the gender of the speaker, etc. The formant features together with the information for each vowel token were stored in a dictionary, and all the dictionaries were saved in a list and serialized in pickle format.

The second feature set consisting of high-level representations of the speech sounds were generated from the HuBERT acoustic model (Hsu et al., 2021), using the existing implementation from the Transformers library in Huggingface (Wolf et al., 2019). The HuBERT model architecture follows the wav2vec 2.0 (Baevski et al., 2020) architecture including a convolutional encoder and a BERT encoder. As a self-supervised speech representation learning, HuBERT is trained on masked prediction tasks and is able to generates hidden states as the internal representation vectors of the input audio. It adopts an offline clustering step to provide aligned target labels for a BERT-like prediction loss (Hsu et al., 2021), and its performance

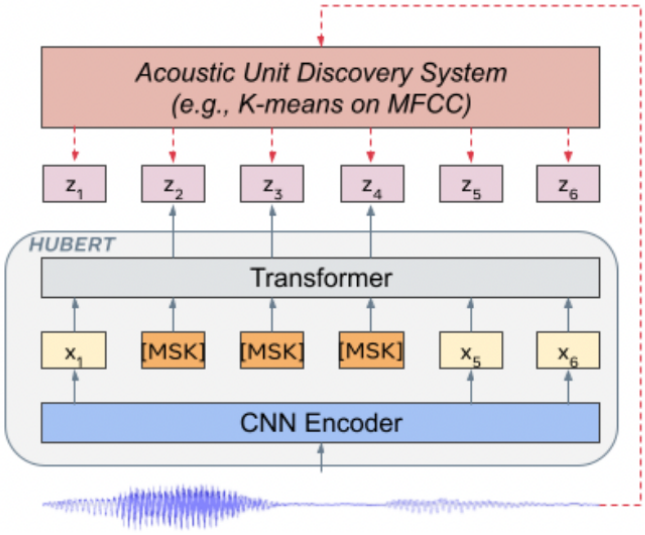

Figure 1. HuBERT architecture (from Hsu et al., 2021, p.2)

either matches or improves upon the performance of wav2vec 2.0-based learning. The current study used audio files as input for HuBERT, and individual hidden units were generated for smaller segments of the audio files, which is of 20 msec each. Based on the time at the midpoint of the target vowel token, I was able to locate the smaller segment that the target vowel token fell into, obtain its hidden units, and store the high-dimensional vector into another dictionary, along with other information about the token (the following nasal token, the word where the vowel occurs, etc.). A list containing dictionaries representing each vowel token is serialized in a second pickle file.

Therefore, the two pickle files each contains a different type of feature vectors of /a/ tokens. Among 1538 tokens in each of the two data files, 1038 was saved as training set, 300 was saved as development set, and 200 as test set.

2.3 Machine Learning Models

Logistic regression model and SVM were compared for the classification task, using the existing implementations from sklearn. Logistic regression is an error-driven model that provides probabilities for each class. A small adjustment is made to all weight vectors for each iteration in the training process. Parameters such as regularization strength can be adjusted to optimize fitting.

The SVM classifier takes the dot product of an example and parameters, and identifies the optimal classification boundary for the training set. It seeks for a separating boundary that maximizes the margin. SVM is able to deal with non-linear decision boundary with a more complex kernel function, such as polynomial and RBF kernel.

In the current task, both the logistic regression and SVM are arguably appropriate and robust for binary classification (i.e., [a] or [ɑ]). Both of them can handle data sets that are linearly separatable, but SVM might have higher accuracy when the data sets have high dimensional space (e.g., the HuBERT-generated hidden states).

2.3.1 Implementations

The two sets of features and the numerically-encoded gold-standard label for each token were separately stored in lists for train, dev, and test data sets. Before implementing the two models, a majority baseline classifier was first implemented using sklearn, which simply outputs the vowel class that occurs most frequently. It was evaluated on the development (dev) data set, with an f-score of 0.39.

I then used each of the two training data sets that contained different types of features to train the two models, with the default parameters in sklearn. The logistic regression model with the acoustic features only leads to an f-score of 0.595 on the dev data set, and with HuBERT-generated features has a much higher f-score of 0.906 on the dev set. Similarly, the SVM classifier with the acoustic features has a pretty low f-score of 0.432 on the dev data set, and with HuBERT-generated features has considerably higher f-score of 0.902 on the dev set.

2.3.2 Fine-tuning the hyperparameters

To improve the f-score, a set of hyperparameters were considered for fine-tuning the models. For the logistic regression model, the training epochs varies in the range of 1 to 20 for the model with acoustic features and of 25 to 35 for the model with HuBERT features. The regularization strength varies in [0.1, 0.05, 0.01, 0.005, 0.001], and regularization type is set to be either “lbfgs” or “liblinear”. For the SVM mode, the training epochs are set in the range of 1 to 20 for the model with acoustic features and of 100 to 500 for the model with HuBERT features. the regularization strength in the range of [0.1, 0.05, 0.01, 0.005, 0.001], kernel is set as either “linear”, “poly” or “rbf”, and gamma is either “scale” or “auto”.

Based on the dev set, the highest f-score for logistic regression model with the acoustic feature set is 0.608, with the best training epochs 6, regularization strength 0.1, and regularization type "lbfgs". The highest f-score for logistic regression model with the HuBERT-generated features is 0.914, with the best training epochs 26, the best regularization strength 0.1, and the best regularization type "lbfgs". The highest f-score for SVM model with the acoustic feature set is 0.557, with the best training epochs 1, the best regularization strength 0.1, and the best kernel "linear", and the best gamma "auto". The highest f-score for SVM model with the HuBERT-generated feature set is 0.922, with the best training epochs 250, the best regularization strength 0.001, and the best kernel "poly", and the best gamma "auto".

3. Evaluation

Using fine-tuned parameters, I compared the performance of the logistic regression classifier and SVM with each of the two feature sets as input on the test data set. The two classifiers are expected to have higher f-scores with the high-dimensional HuBERT-generated feature vectors as input than the acoustic formant features. Indeed, the f-score of logistic regression with acoustic features is 0.538, considerably lower than that with the HuBERT-generated hidden states features, which is of 0.850 f-score. Similarly, SVM with HuBERT-generated features has an f-score of 0.866, a lot higher than the f-score of 0.625 of SVM with acoustic features. The SVM classifier with either feature set has a slightly better performance on the test set than logistic regression.

Significance testing was conducted to see whether there is a difference between the two classifiers using each of the feature set. Specifically, three t-tests were run to test the following null hypotheses (1) there is no difference between logstic regressions with the two feature sets; (2) there is no difference between SVMs with the two feature sets; (3) there is no difference between logistic regression and SVM with HuBERT-generated feature set. The p-value of test (1) is 0.0015 which is less than 0.5%, so we can reject the first null hypothesis and say that there is a significant difference between logstic regressions with the two feature sets. The p-value of test (2) is 0.0007 which is less than 0.5%, so we can reject the second null hypothesis and say that there is a significant difference between SVMs with the two feature sets. The p-value of test (3) is 0.250 which is not less than 0.5%, so we cannot reject the third null hypothesis. There is no significant difference between logistic regression and SVM with HuBERT-generated features.

4. Results and Discussions

Table 1 presents the key results illustrated in the evaluation section, including the f-scores of the two models with different feature sets, and the significance testing between models and features.

Model	Feature set	F-score	t-test	
Logistic regression	Acoustic formants	0.538	$p < 0.5\%$	$p = 0.250$ with HUBERT
	HuBERT	0.850		
SVM	Acoustic formants	0.625	$p < 0.5\%$	
	HuBERT	0.866		

Table 1. Comparison between models with the two feature sets based on f-scores and t-tests.

In order to visualize the internal structure of the data, UMAP are used to plot the HuBERT-generated hidden states of the two vowel classes in high-dimensional space and project the graph into 2D. The UMAP algorithm was implemented from the UMAP library (McInnes & Healy, 2018). The number of neighbors was set in the range of 2 to 500, and the minimum distance between points was 0.2, with the metric "Euclidean". As seen from Figure 2, the cluster structure of [a] and [ɑ] becomes a bit clearer when the number of neighbor increases. Nevertheless, there are still lots of overlap in the two sets of feature vectors, suggesting some level of similarities between the extracted hidden states of the two allophones.

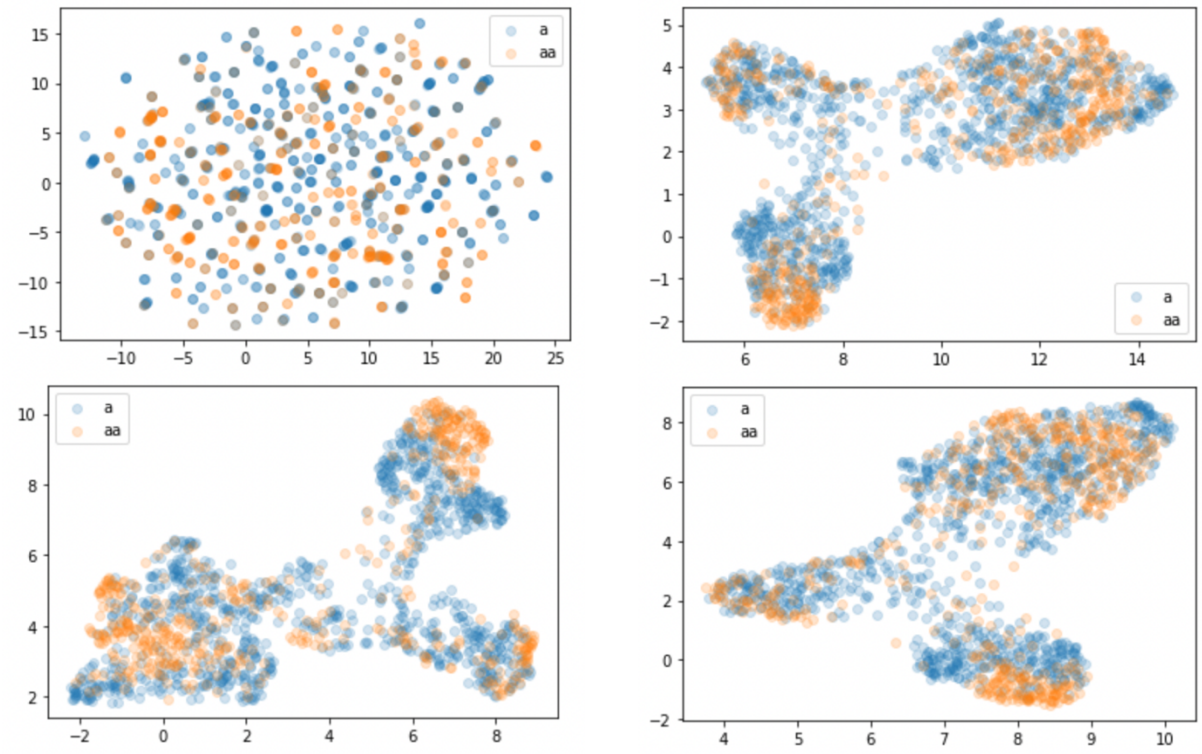

Figure 2. Internal structures of HuBERT-generated feature vectors of [a] (before [n], coded as "a") and [ɑ] (before [ŋ], coded as "aa"). The number of neighbors is 2 (top left), 20 (bottom left), 200 (top right) and 500 (bottom right).

Classification errors generated by the two models in the test set were analyzed. Specifically, the phone that preceded the target vowel token was examined, given a potential coarticulatory effect on the pronunciation of the vowel. It was found that most of the error tokens were preceded by glides /j, w/ (transcribed as “y” and”w”) or retroflex affricates /ʈ͡ʂ, ʈ͡ʂʰ/ (transcribed as “zh” and “ch”). Both the glides and the retroflex affricates can cause articulatory and acoustic effects on the following vowel, regardless of what nasal it precedes. In particular, several tokens of the front back vowel [ɑ] were misclassified as the front vowel after the glide /j/ by the SVM model with the HuBERT features. This is possibly because the relatively anterior articulatory gesture of /j/ had a coarticulatory effect on the following vowel, so that the vowel was produced a bit more forward that it normally would.

The speech rate of the relevant utterance where the token belongs to could also influence the performance. Faster speech rate can possibly lead to more errors due to phonetic reductions. The speech style, namely, read speech and spontaneous speech might also have an effect. Although a few errors tokens have extremely fast speech rate (about 10 syllables/second), most of the error tokens fall in the range of 3-7 syllables/sec. The speech style also does not seem to have a consistent effect on the accuracy. Misclassified tokens were quietly evenly from both speech styles.

References:

Almekhlafi, E., Moeen, A. M., Zhang, E., Wang, J., & Peng, J. (2022). A classification benchmark for Arabic alphabet phonemes with diacritics in deep neural networks. *Computer Speech & Language, 71*, 101274.

Baevski, Alexei, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations. (2020).

Bissell, M. (2021). Automatic phonetic classification of vocalic allophones in Tol. *Proceedings of the Linguistic Society of America*, *6*(1), 403-410.

Boersma, Paul & David Weenink. (2016). *Praat: doing phonetics by computer.* http://www.praat.org/.

Chuang, C. T. (2017). Revisiting Nasal Coda Merger in Taiwan Mandarin: A Corpus Study. *Concentric: Studies in Linguistics*, *43*(2).

Duanmu, S. (2007). *The phonology of standard Chinese*. OUP Oxford.

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.

McInnes, L.& Healy, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv*: 1802.03426.

Mukherjee, H., Dutta, M., Obaidullah, S. M., Santosh, K. C., Gonçalves, T., Phadikar, S., & Roy, K. (2018). Performance of Classifiers on MFCC-Based Phoneme Recognition for Language Identification. In *International Conference on Computational Intelligence, Communications, and Business Analytics* (pp. 16-26). Springer, Singapore.

Pedregosa F., et al. (2011). Scikit-learn: Machine Learning in Python. *JMLR 12*. 2825-2830.

Piotrowska, M., Korvel, G., Kostek, B., Ciszewski, T., & Czyżewski, A. (2019). Machine learning-based analysis of English lateral allophones. *International Journal of Applied Mathematics and Computer Science*, *29*(2).

Wolf, T. et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv:*1910.03771.

Yousafzai, J., Ager, M., Cvetkovic, Z., & Sollich, P. (2008). Discriminative and generative machine learning approaches towards robust phoneme classification. In *2008 Information Theory and Applications Workshop* (pp. 471-475). IEEE.