ECE 214A Project Report

Jiusi Zheng[1], *Zilai Wang*[1], *Kangrui Chen*[1]

[1]UCLA, USA

zilaiwang2001@ucla.edu, zheng94@g.ucla.edu, kangrui@g.ucla.edu

Abstract

In this report we will implement a speaker region identification system that predicts the city of origin of the speaker of a given utterance. Methods including data augmentation, neural network-based encoders, and spectral subtraction are used to perform the task. During the blind test, the spec augment method demonstrates 71% hidden clean accuracy and 69% hidden noisy accuracy; the DNN encoder achieves 81% hidden clean accuracy and 75% hidden noisy accuracy.

Index Terms: Data augmentation, Encoders, Spectral subtraction

1. Introduction

Understanding the geographical origin of speakers based on their speech signals is vital across forensic linguistics, sociolinguistics, and speech technology applications. Implementing a speaker region identification system to predict the city of origin from a given utterance is crucial, benefiting various domains.

This paper explores implementing a speaker region identification system using acoustic features such as MFCCs, DNN encoder, spectral subtraction, and data augmentation methods like spec augment. We investigate its applications, discuss technical challenges, and present experimental results to demonstrate effectiveness in identifying the city of origin of speakers based on speech signals.

2. Background

2.1. Data Augmentation

Data augmentation has emerged as a promising strategy for enhancing training datasets in Automatic Speech Recognition (ASR). This technique, exemplified by various studies [1, 2], entails the generation of synthetic data to augment sparse resources in speech recognition tasks.

SpecAugment is a novel augmentation method that targets the log mel-spectrogram of input audio instead of manipulating the raw audio directly [3]. This approach is distinguished by its simplicity and low computational demand, as it treats the log mel spectrogram akin to an image, obviating the need for supplementary data. Consequently, SpecAugment can be seamlessly integrated into training processes. The technique encompasses three distinct spectrogram deformations: time warping, which alters the time series along the time axis; and time and frequency masking.

Despite its elementary nature, SpecAugment has demonstrated remarkable efficacy, enabling the training of end-to-end ASR networks to outperform complex hybrid systems and set new benchmarks, even in the absence of Language Models (LMs).

2.2. Weak Supervision and Encoders

2.2.1. Weak Supervision

Deep learning methods like self-supervised learning [4] are widely used in speech tasks such as Automatic Speech Recognition (ASR) [5], Sentiment Analysis [6], and dialect classification [7]. However, limited datasets pose a challenge, particularly in tasks like dialect detection based on African American English (AAE) [8]. Weak supervision has proven crucial in low-resource scenarios [9]. Acoustic features from models like Wav2vec2 and X-vector are effective in dialect-related tasks [10], yet training large language models remains costly and time-consuming. Our approach combines traditional speech features with shallow networks, achieving high accuracy, especially in noise-free conditions. We employ various shallow neural network architectures as encoders for feature extraction in dialect classification. During training, we use traditional acoustic features and labels as input, with the index of the maximal value serving as the predicted label for evaluation.

2.2.2. Encoders

Encoder is a fundamental component of neural network architectures which has been extensively utilized within transformer-based model architectures like Wav2vec2 [11] and Hubert [12] self-supervised learning models. The primary function of an encoder is to transform input data into a latent representation, often of lower dimensionality, that captures the salient features or characteristics of the input data. This process involves passing the input through a series of layers, typically consisting of various types of neural network units such as convolutional layers, recurrent layers, or fully connected layers. The encoder's output, known as the encoded representation or embedding, serves as a compressed and abstract representation of the input data, which can be subsequently used for downstream tasks like dialect classification in this work.

2.3. Spectral Subtraction

The main idea is to estimate noise spectrum from speech pauses, which is then subtracted from the total noisy speech spectrum to estimate the clean speech. Speech and noise are assumed to be uncorrelated, additive, and stationary. In late 1970s and 1980s, spectral subtraction [13] was proposed as one of the first effective techniques for noise reduction in various speech signals. Over the years, entering 2000s, researchers have continued to refine spectral subtraction and develop more robust noise reduction algorithms. Techniques such as Wiener filtering [14], minimum mean square error (MMSE) estimation [15], and adaptive filtering [16] have been introduced to address the limitations of spectral subtraction.

In Fourier domain the spectral subtraction method can be summarized as follows,

$$
\begin{aligned}
|\hat{S}(\omega)|^2 &= |Y(\omega)|^2 - |\hat{D}(\omega)|^2 && (1)\\
&= (1 - \frac{|\hat{D}(\omega)|^2}{|Y(\omega)|^2}) * |Y(\omega)|^2 && (2)\\
&= H(\omega)^2 |Y(\omega)|^2 && (3)
\end{aligned}
$$

where step (2) is a simple factorization and step (3) defines the spectral subtraction filter (SSF). The remaining phase information is not as important for speech analysis, thus is ignored in this project. Many other variant methods tend to make additional assumptions about the the signal quality or the type of noise, which may not hold for the dataset used in this project.

3. Project Description

This section provides an overview of the project including the methodologies and technologies employed.

3.1. Spec_Augmentation

3.1.1. Implementation

This subsection outlines the data augmentation methods employed in this study. Spec_Augmentation involves three key steps: time warping, frequency masking, and time masking. The implementation process is shown in Figure 1.

Time warping involves randomly selecting a point in the spectrogram or time-frequency representation of an audio signal and shifting subsequent time frames forward or backward by a randomized interval. This process modifies the temporal scale while preserving pitch and other characteristics. Warping parameters are typically constrained within predefined limits for realism and relevance.

Frequency masking, unlike time warping, alters the spectral content of audio signals. It simulates situations where certain frequencies are obscured by louder neighboring frequencies, emulating natural auditory phenomena.

Time masking aims to enhance model robustness by reducing sensitivity to temporal variations in audio signals. It simulates instances where parts of an audio signal are temporarily obscured by overlapping sounds, interruptions, or background noise. Training on data with time masking instances enables models to prioritize persistent features over time, improving pattern discernment and information extraction across varied conditions.

In this approach, we only augmented the Class LES which labeled as 1.

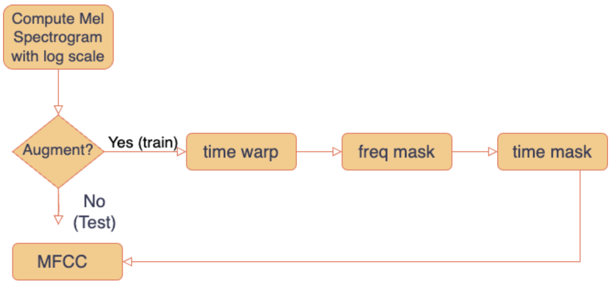


Figure 1: *Spec Augment*

3.1.2. Features and parameters

For fairness compared with the baseline, we choose the sampling rate as 44.1 KHz and use MFCC 13 as the method of feature extraction after applying Spec_Augmentation. The hyperparameters of time warping W, frequency masking parameter f para, the time masking parameter t para, frequency mask number f num, time mask number t num are shown in Table 1.

Table 1: *Spec_Augment parameters*

W	f para	t para	f num	t num
20	15	20	1	1

3.1.3. Results

Table 2: *Spec_Augment*

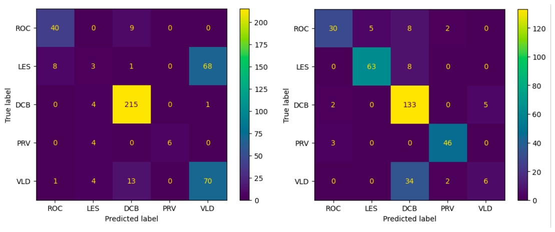
Method	run time	Test Clean acc	Test Noisy acc
Baseline	10 mins	76.9%	61.9%
Spec_Augment w/ time warp	12 mins	74.3-77.9%	74.1-82.4%
Spec_Augment w/o time warp	11 mins	74.7-77.2%	73-81%

Table 3: *Class Augmented*

Class augmented	Test Clean acc	Test Noisy acc
Baseline	76.9%	61.9%
LES	74.3-77.9%	74.1-82.4%
LES and VLD	75.1-76.9%	73.7-80.7%

Table 4: *Blind test results*

Blind test	Hidden Clean acc	Hidden Noisy acc
Spec Augment	71%	69%



(a) *Spec Augment test clean (class 1 LES augmented)* (b) *Spec Augment test noise (class 1 LES augmented)*

Figure 2: *Comparison of clean and noisy test results*

3.2. Encoders for Feature Extraction

The acoustic features related to dialect detection include different pronunciations, vocabulary usage, and grammar. Pronunciation and vocabulary usage are features that are independent of temporal order, while grammar is a feature that is dependent on temporal order. Based on these potential features for dialect classification, we employed different architectures to detect different features. In this work, we utilized both Deep Neural Network(DNN) and multi-head self-attention network without masking mechanism to detect features independent of temporal order. Subsequently, we employed RNN networks based on long short time memory(LSTM) to detect features dependent on temporal order. The features extracted by the encoders we used are based on traditional acoustic features and in this work we focused on Mel-Frequency Cepstral Coefficients and Perceptual Linear Prediction features. The overall workflow is illustrated in the following diagram.

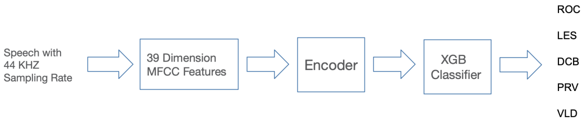


Figure 3: *Overview of Encoder Feature Extraction*

Deep Neural Network (DNN) structures have proven valuable in classification tasks, while multi-head self-attention mechanisms, prominent in Transformers, are efficient, especially in self-supervised learning. In our study, we aimed to compare their efficacy in low-resource environments for dialect classification. We implemented two innovative approaches in DNN and Self-Attention encoders. Firstly, we computed MFCC features for every 10 or 100 frames of audio files based on their lengths, enabling the encoder to learn phoneme-level acoustic features, resulting in improved classification accuracy in clean environments (DNN: 91.1%, Self-Attention: 89.2%). In noisy environments, DNN and self-attention encoders achieved accuracies of 65.4% and 62.8%, respectively. Additionally, we observed that the encoder acted as a denoiser, improving classification accuracy by approximately 5% in noisy tests and 15% in noisy blind tests. Recognizing noise characteristics in 39-dimensional MFCC features, we reduced them to 5 dimensions during encoding, enhancing accuracy in noisy environments. Despite encountering various noise types in the dataset, we have yet to identify a more suitable filter for speech enhancement within our time constraints.

In feature selection, we explored MFCC (Mel-Frequency Cepstral Coefficients) and PLP (Perceptual Linear Prediction) features. MFCC features represent the short-term power spectrum of a sound, mimicking the human auditory system's response. PLP features, derived from LPC (Linear Predictive Coding) spectrum with additional processing, approximate human auditory perception. However, the encoder struggled to extract useful information from the 39-dimensional PLP features. Thus, we chose to use MFCC features for training the encoder.

Among these, MFCC features notably enhanced classification accuracy, especially in noise-free conditions, achieving a peak accuracy of 96% on our current test dataset. However, we observed fluctuations in accuracy during 30 test evaluations, with variations of up to 5%. To address potential overfitting and ensure consistent performance on unknown datasets, we adjusted parameters to limit accuracy fluctuations within 1% over 30 tests. Although the accuracy slightly decreased to 91.2% in clean conditions compared to the previous best of 96%, it remained more stable. Figure 4 displays the confusion matrices of the XGB classifier using features from the DNN encoder, while Figure 5 illustrates the training and validation accuracy curves for the three different encoders.

3.2.1. *Long Short Time Memory Encoder*

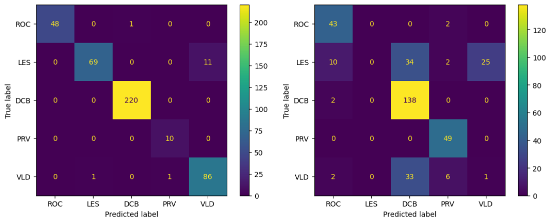
Long Short-Term Memory (LSTM) networks, a variant of recurrent neural networks (RNNs), excel at capturing long-term dependencies in sequential data, making them ideal for tasks like time series prediction. Their intricate memory cell structure enables retention of information over extended periods, enhancing their efficacy in time-related tasks. In our study, we leverage LSTM to capture time-related features such as grammar in speech for dialect classification. Employing a 6-layer LSTM neural network, we aim to extract time-dependent information, operating under the assumption that, as per LSTM principles, the last feature vector in the output sequence contains the most informative details. For detailed comparison, the results of different encoders are presented in Table 4.

Table 5: *Classification Accuracy for Each Encoder*

Method	Training accuracy	Test Clean accuracy	Test Noisy accuracy
DNN	100%	91.1%	65.5%
Self-attention	100%	89.2%	62.8%
LSTM	98.4%	87.6%	57.6%
Baseline	100%	76.9%	61.9%

Table 6: *Blind test results*

Blind test	**Hidden Clean acc**	**Hidden Noisy acc**
DNN Encoder	81%	75%



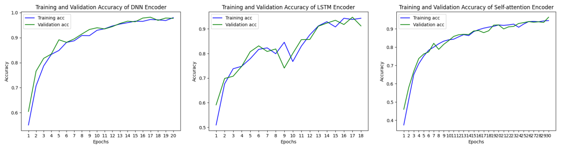
(a) *Confusion matrix in clean* (b) *Confusion matrix in noisy*

Figure 4: *Comparison of clean and noisy test results*

3.3. Spectral Subtraction

3.3.1. *implementation*

The spectrum for the total speech signal and the estimated noise are first extracted using the public available librosa STFT (short-time Fourier transform) library. The noise spectrum estimation algorithm is developed from scratch and is introduced in the following paragraphs. For each frequency where the noisy



(a) *DNN* (b) *LSTM* (c) *Self-attention*

Figure 5: *Training and Validation Accuracy Curve*

signal spectrum magnitude is less than the average, the output is set to 0. Since the speech and the part of the noise lying above the average remain after this step, residual noise reduction is employed afterwards.

The implementation of the noise spectrum aims to provide as much filtering as possible without compensating computational efficiency. The algorithm default starts by searching a continuous speech segment of pure noise of a certain width, which is a manually tuned hyperparameter. Thereafter, the speech length is normalized with respect to this new metric of unit time and the remaining is chopped off. A major hyperparameter to be tuned is the scaling factor with which the noise spectrum is subtracted from the entire spectrum. After finishing the previous steps, the spectrum is converted back to time domain via inverse STFT and the output of the algorithms is an approximate clean signal that is going to be subjected to feature extraction methods. The algorithm is compatible with the conventional feature extraction methods.

Table 7 summarizes the performance of the spectral subtraction method integrated with some feature extraction methods. Only the best-performing models are reported here. As shown in the table, traditional 13-coefficient MFCC and mel-spectrogram features bring the best performance on both clean and noisy dataset when combined with SS, with the clean and noisy test error sitting at percentages of 78.7 and 64.8, respectively. The RobustScalar function under the Python preprocessing library is utilized to normalize the feature vectors and also to eliminate the effects of data outliers. The selected and normalized features are concatenated and averaged as the final one-dimensional output. Additional attempts had been made to tune other hyperparameters in the pipeline like the XGB classifier parameters (via grid search), but the performance improvement by parameter tuning itself usually caps at around 2%, for both clean and noisy test data. The mechanism of the classifier is not well-understood and is not within the scope of the project.

Table 7: *Performance of spectral subtraction*

Method	Training accuracy	Test Clean accuracy	Test Noisy accuracy
Baseline	100%	76.9%	61.9%
MFCC + Resampling	100%	74.8%	63.5%
SS + MFCC	100%	89.2%	62.8%
SS + MFCC + Resampling	100%	86.1%	59.2%
SS + MFCC + Melspectrogram	100%	78.7%	64.8%
Baseline	100%	76.9%	61.9%

4. Summary and Discussion

In this section, we will discuss the results of our models and ideas for future work.

4.1. Spec Augment

Table 2 illustrates that SpecAugment significantly enhances the noisy test accuracy by 13% to 20%, with negligible changes in clean test accuracy, indicating its efficacy in accent discrimination under noisy conditions. This phenomenon implies that SpecAugment's augmented data more closely matches the features of the noisy test set than the clean one.

As per the insights from [3], while time warping positively impacts model performance, its absence does not markedly reduce the efficacy of SpecAugment, underscoring its non-critical role in performance gains.

Further, Figure 4b reveals targeted training augmentation of the LES accent significantly improves classification in noisy conditions. In contrast, augmenting both LES and PRV classes doesn't yield superior outcomes than augmenting the LES class alone, suggesting that augmenting the VLD class does not enhance its recognition in noisy tests. This could be attributed to the disparity between the augmented VLD training data and the characteristics of the VLD noisy test dataset.

4.2. Encoders

After comparison, we found that the Deep Neural Network (DNN)-based encoder currently outperforms other encoders in dialect classification on the existing dataset, regardless of noise levels. We attempted to improve classification accuracy by combining data augmentation with the DNN encoder, but the combined effect was not significant. The XGB classifier achieved an accuracy of 72.7% in noise-free environments and 67.4% in noisy environments after combining the two methods. We observed that data augmentation significantly improves accuracy in noisy environments. Hence, a potential research direction is to develop a suitable filter or data augmentation method for the DNN encoder to further enhance accuracy in noisy environments.

4.3. Spectral Subtraction

The spectral subtraction techniques have under-expected performance possibly due to the over-generalization of spectrum properties. Also, the SS seems to corrupt the clean signals and decrease clean speech performance by a percentage of around 5 to 10. However, the spectral subtraction method still outperforms less-generalized methods like making specific assumptions on the noise type. A comparison dataset is degraded by white Gaussian noise (AWGN) stationary noise, at four SNR levels: 0 dB, 5 dB, 10 dB, and 15 dB, implemented as presented in one of the previous studies [17]. The best-performing results at 5dB corruption is lower than the baseline performance and thus not reported here.

The integration of the spectral subtraction method with the best-performing encoder degrades the performance of both methods, thus the results are omitted in the report.

5. References

[1] N. Kanda, R. Takeda, and Y. Obuchi, “Elastic spectral distortion for low resource speech recognition with deep neural networks,” in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 309–314.

[2] A. Ragni, K. M. Knill, S. P. Rath, and M. J. Gales, “Data augmentation for low resource languages,” in *INTERSPEECH 2014: 15th annual conference of the international speech communication association*. International Speech Communication Association (ISCA), 2014, pp. 810–814.

[3] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” *arXiv preprint arXiv:1904.08779*, 2019.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[5] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Applying wav2vec2. 0 to speech recognition in various low-resource languages,” *arXiv preprint arXiv:2012.12121*, 2020.

[6] Y. Zhang, B. Xu, and T. Zhao, “Convolutional multi-head self-attention on memory for aspect sentiment classification,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1038–1044, 2020.

[7] L. Lulu and A. Elnagar, “Automatic arabic dialect classification using deep learning models,” *Procedia computer science*, vol. 142, pp. 262–269, 2018.

[8] S. S. Mufwene, J. R. Rickford, G. Bailey, and J. Baugh, *African-American English: structure, history, and use*. Routledge, 2021.

[9] W.-N. Hsu *et al.*, “Speech processing with less supervision: learning from weak labels and multiple modalities,” Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[10] A. Johnson, K. Everson, V. Ravi, A. Gladney, M. Ostendorf, and A. Alwan, “Automatic dialect density estimation for african american english,” *arXiv preprint arXiv:2204.00967*, 2022.

[11] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[12] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[13] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[14] V. Stahl, A. Fischer, and R. Bippus, “Quantile based noise estimation for spectral subtraction and wiener filtering,” vol. 3, pp. 1875–1878 vol.3, 2000.

[15] V. K. Gupta, A. Bhowmick, M. Chandra, and S. N. Sharan, “Speech enhancement using mmse estimation and spectral subtraction methods,” pp. 1–5, 2011.

[16] R. Martinez, A. Alvarez, P. Gomez, V. Nieto, and V. Rodellar, “Combination of adaptive filtering and spectral subtraction for noise removal,” vol. 2, pp. 793–796 vol. 2, 2001.

[17] N. Upadhyay and A. Karmakar, “Speech enhancement using spectral subtraction-type algorithms: A comparison and simulation study,” *Procedia Computer Science*, vol. 54, pp. 574–584, 2015.