Using Computational Models to Study Accent Adaptation: A Tutorial

Samantha Chiu[2*], Leo Moore[1*], Zoey Liu[3] & Ethan Kutlu[1,2]
[1]Department of Linguistics, [2]Department of Psychological and Brain Sciences, University of Iowa
[3]Department of Linguistics, University of Florida

VOICE LAB
Variability of identity through context and experience

Computational modeling of speech recognition offers a promising tool to study human speech perception.

INTRO

Humans extract acoustic features from audio to identify phonemes to decode words.

We can simulate this process with machine learning (neural networks).

Pytorch contains off-the-shelf neural networks trained to do tasks like speech recognition.

Backbone: neural network that contains the mapping from audio to phoneme.

Decoder: algorithm that decides the phoneme based on the audio.

Automatic Speech Recognition (ASR): neural networks trained to convert audio into text

Word Error Rate (Levenshtein Distance): difference between two texts/strings.

METHODS (adapted from Hira, M., 2024)

Get Audio → Clean all files (remove noise, extra words) → Match sampling rate to PyTorch requirements

Load PyTorch

```
import torch
import torchaudio
import re
import numpy as np
#set seed
torch.random.manual_seed(0)
#determine how to run the algorithm (i.e. on your CPU or through CUDA)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

Load backbone & decoder

```
#define decoder algorithm
class GreedyCTCDecoder(torch.nn.Module):
    def __init__(self, labels, blank=0):
        super().__init__()
        self.labels = labels
        self.blank = blank

    def forward(self, emission: torch.Tensor) -> str:
        indices = torch.argmax(emission, dim=-1)  # [num_seq,]
        indices = torch.unique_consecutive(indices, dim=-1)
        indices = [i for i in indices if i != self.blank]
        return "".join([self.labels[i] for i in indices])
#set backbone
bundle = torchaudio.pipelines.WAV2VEC2_ASR_BASE_960H
model = bundle.get_model().to(device)

decoder = GreedyCTCDecoder(labels=bundle.get_labels())
```

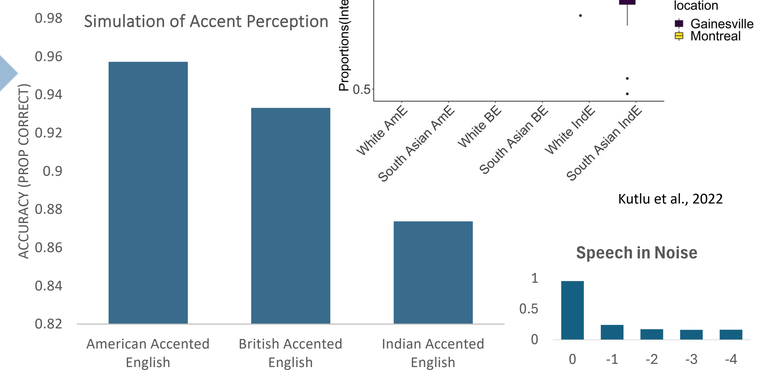
Run ASR

```
waveform, sr = torchaudio.load(currfpath + i)
waveform = waveform.to(device)

with torch.inference_mode():
    emission, _ = model(waveform)

transcript = decoder(emission[0])
transcript = re.sub("\\|", " ", transcript).rstrip()
results.append(transcript)
```

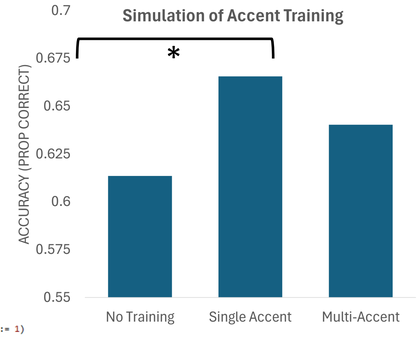
Calculate Word Error Rate

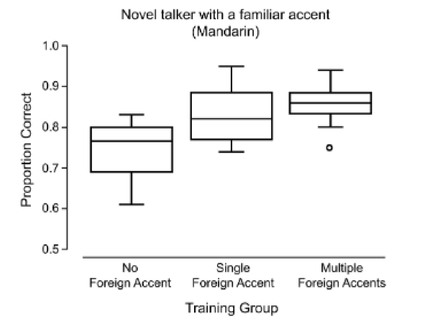
```
#%% Levenshtein distance
def word_error(correct, guess):
    matrix = np.zeros((len(guess)+1, len(correct)+1))
    for i in range(1, len(guess)+1):
        matrix[i,0] = i
    for j in range(1, len(correct)+1):
        matrix[0,j] = j
    for j in range(1,len(correct)+1):
        for i in range(1,len(guess)+1):
            (isdiff := 0) if correct[j-1] == guess[i-1] else (isdiff := 1)
            matrix[i,j] = min(matrix[i-1, j] + 1,          # Deletion
                              matrix[i, j-1] + 1,          # Insertion
                              matrix[i-1,j-1] + isdiff)    # Substitution
    return matrix[len(guess), len(correct)] / max(len(correct), len(guess))
score = word_error(correct[num], transcript)
```

RESULTS



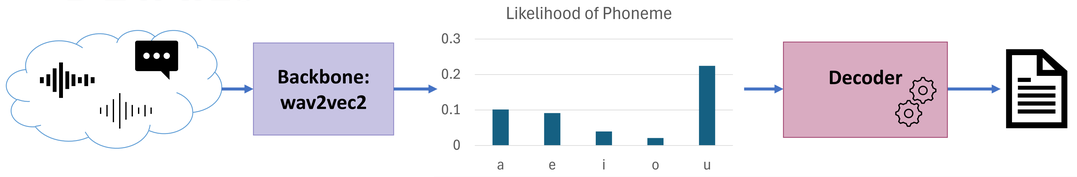
Kutlu et al., 2022



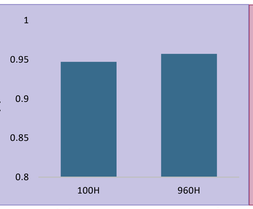


Baese-Berk, Bradlow, Wright, 2013

DETAILS



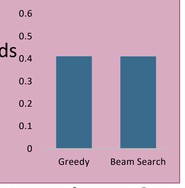
Computational models can be trained on different sized datasets.



Types of Decoders (non-inclusive):

1. Greedy: Pick the most likely option at each decision
2. Beam Search: Pick the most likely option over n words

The cat	... dashed	... to the	... door.
50%	40%	45%	80%
The car	... crashed	... to the	... door.
50%	60%	55%	20%



Word Error Rate: How many changes need to be made to match the original text?

Types of Changes:
1. Addition
2. Insertions
3. Substitutions

Response	_	L	U	C	K	R	E	C	E	I	P	T	Y	O	U	_	O	U	R
Answer	S	L	A	C	K	R	E	C	I	E	P	T	E	W	E	H	O	U	R

Word Error Rate

(1 insertion + 1 substitution) ÷ 5 characters = 40% incorrect (60% correct)

This measure is imperfect; this mistake would count as 2 errors instead of 1 (swap E and I)

Scoring homophones causes issues (3/3 errors and 1/4 errors)

FUTURE DIRECTIONS

- Validate automatic speech recognition models as tools for human speech perception
- Simulate learning/training studies via fine-tuning (i.e. fine-tune an English ASR model with Spanish to simulate Spanish language learning)

REFERENCES

Hira, M., (2024) . Speech Recognition with wav2vec2. PyTorch. https://pytorch.org/audio/stable/tutorials/speech_recognition_pipeline_tutorial.html
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.
Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., & Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (pp. 929–947). Association for Computing Machinery.
Radford, A., Kim, J., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org. Available from https://proceedings.mlr.press/v202/radford23a.html.
Baese-Berk, M. M., Bradlow, A. R., & Wright, B. A. (2013). Accent-independent adaptation to foreign accented speech. *The Journal of the acoustical society of America*, *133*(3), EL174-EL180.
Kutlu, E., Tiv, M., Wulff, S., & Titone, D. (2022). Does race impact speech perception? An account of accented speech in two different multilingual locales. *Cognitive Research: Principles and Implications*, 7(1), 7.

*Equal Contributors

IOWA