Evaluating Automatic Pronunciation Scoring with Crowd-sourced Speech Corpus Annotations

Nils Hjortnæs, Daniel Dakota, Sandra Kübler, Francis Tyers
Indiana University
{nhjortn, ddakota, skuebler, ftyers}@iu.edu

Abstract

Pronunciation is an important, and difficult aspect of learning a language. Providing feedback to learners automatically can help train pronunciation, but training a model to do so requires corpora annotated for mispronunciation. Such corpora are rare. We investigate the potential of using the crowdsourced annotations included in Common Voice to indicate mispronunciation. We evaluate the quality of ASR generated goodness of pronunciation scores through the Common Voice corpus against a simple baseline. These scores allow us to see how the Common Voice annotations behave in a real use scenario. We also take a qualitative approach to analyzing the corpus and show that the crowdsourced annotations are a poor substitute for mispronunciation annotations as they typically reflect issues in audio quality or misreadings instead of mispronunciation.

1 Introduction

Pronunciation of utterances is a difficult task for language learners, and there is limited research on how best to generate feedback automatically (Agarwal and Chakraborty, 2019; Moses et al., 2020; Neri et al., 2006; Witt, 2012). However, such feedback can be an invaluable tool for those learning a language who want to improve their speaking skills, allowing them to practice when a human teacher is not available. Ideally, the feedback should reflect the judgements of a native speaker of the targeted language variant and be targeted at the learner's desired dialect (e.g., British vs. American English) and skill level. One current method for evaluating pronunciation is to interpret the confidence of an Automatic Speech Recognition (ASR) model as the goodness of pronunciation (Moses et al., 2020). Doing so makes a crucial assumption that the accuracy of the transcription is representative of the learner's pronunciation accuracy.

One of the challenges in investigating the quality of automatic feedback is that there is only one publicly available corpus with human judgements on pronunciation, L2-ARCTIC (Zhao et al., 2018). Since it does not contain examples of native speakers producing the same sentences, we cannot use it for our purposes.

The Common Voice corpus (Ardila et al., 2020) does not contain pronunciation annotation, but does contain upvote and downvote scores per utterance. We propose using these crowdsourced up- and downvote scores as a stand-in for pronunciation scores. We hypothesize that a clip receiving both up- and downvotes indicates a mispronunciation because annotators disagree on the quality, and clips with only upvotes indicate proper pronunciation as well as clear audio. To test whether these labels can be used for evaluating pronunciation scorers, we create a task to classify whether a given audio clip in the Common Voice corpus has any downvotes using the generated pronunciation scores as input. Assuming that the output of a Speech Recognition model is a measure of pronunciation accuracy (Moses et al., 2020), a neural model should be able to use that output to predict the presence of downvotes.

Typically, in ASR, the task is transcribing audio data into orthographic text. In this work we perform a zero-shot classification of downvoted clips using an ASR model (section 5.1). The final layer of this architecture is a softmax layer, providing probabilities, which form the basis of our baseline pronunciation scorer and which we compare across speakers to generate feedback (section 4).

Our results show that detecting downvotes in Common Voice is difficult. The baseline, interpreting the speech recognition softmax output as feedback, achieves only 81.4% with tuning, and in the low 60s when comparing learner's utterances to expert's and predicting downvotes from the comparison. Looking closely at some of the examples and contents affirms that the voting on Common Voice utterances is a poor substitute for mispronunciation annotation. This highlights the need for a dedicated corpus annotated specifically for pronunciation for the development of tools providing pronunciation feedback to language learners.

2 Related Work

Pronunciation feedback systems were researched in depth in the 1990s and 2000s (Witt, 2012), as they have been shown to improve learner's pronunciation (e.g., Agarwal and Chakraborty, 2019; Neri et al., 2006; Dalby and Kewley-Port, 1999). Early pronunciation feedback used Hidden Markov Models (HHMs; Franco et al., 2000; Dalby and Kewley-Port, 1999), following the use of HMMs for Speech Recognition at the time (Malik et al., 2021). Bratt et al. (1998) collected a corpus annotated for pronunciation during this time for evaluating these systems, but it is no longer available.

As speech recognition moved to neural network models (Malik et al., 2021; Hannun et al., 2014), pronunciation feedback followed (Agarwal and Chakraborty, 2019; Moses et al., 2020). Moses et al. (2020) use DeepSpeech (Hannun et al., 2014) to score pronunciation of Te reo Māori, an indigenous language in New Zealand, using their own speech and text corpora by calculating confidence scores for characters, as opposed to utterances, in an elicited sentence or phrase. There is no information available on how the scoring is performed. It appears to consist of the probability of the character from the target sentence appearing at its aligned timestamp, which is interpreted as the model's confidence for that character. They "observed the model working with confident te reo speakers as expected". (Moses et al., 2020)[1]

There are currently many proprietary apps for language learning which include pronunciation training in some form (Coulange, 2023). Common practice for these apps is to give the learner an elicitation phrase and an example of an expert pronouncing it, then request the learner say the phrase. Most apps, such as Memrise[2] and DuoLingo[3], give only binary feedback (correct or incorrect), on a phrase or word level. ELSA[4] is able to give feedback on specific letters, based on phonemes, but only teaches English. Our long term goal is to generate feedback as narrowly as ELSA with a system that can generalize to multiple languages.

3 The Common Voice Dataset

We use the Common Voice English data. Common Voice is a large multilingual collection of audio data for speech recognition crowdsourced by Mozilla (Ardila et al., 2020). It consists of around 1.6 million clips ($\leq$10 sec.) of read sentences/phrases totalling 2 319 hours. Users can contribute recordings of sentence readings, or judgements of other's readings by upvoting or downvoting clips[5]. Only clips with at least one upvote are ultimately included in the validated dataset.

Though the upvotes and downvotes do not necessarily indicate a mispronunciation, they do indicate problems as judged by human contributors. Because mispronunciation is a potential reason for an annotator to downvote a clip, these judgements give us the best indication for which clips are mispronounced.

4 System Overview

4.1 System Pipeline

The pipeline for the process of generating feedback for a given elicited phrase begins with running both the expert and the learner productions of the phrase through the speech recognizer, Coqui (see Section 5.1) and retrieving a softmax probability distribution per time slice. Coqui operates by segmenting an audio file and predicting the character, or lack of a character, present in each segment. This takes the form of a probability distribution over the candidate alphabet. It then recombines the segments into orthography, combining repeating characters[6] and inserting spaces as informed by a language model. Figure 1 shows this process, starting with the extraction of probability distributions in the first transition from the

[1] Only a poster is available for this work https://papareo.nz/docs/PapaReo_NeurIPS2020_Poster.pdf

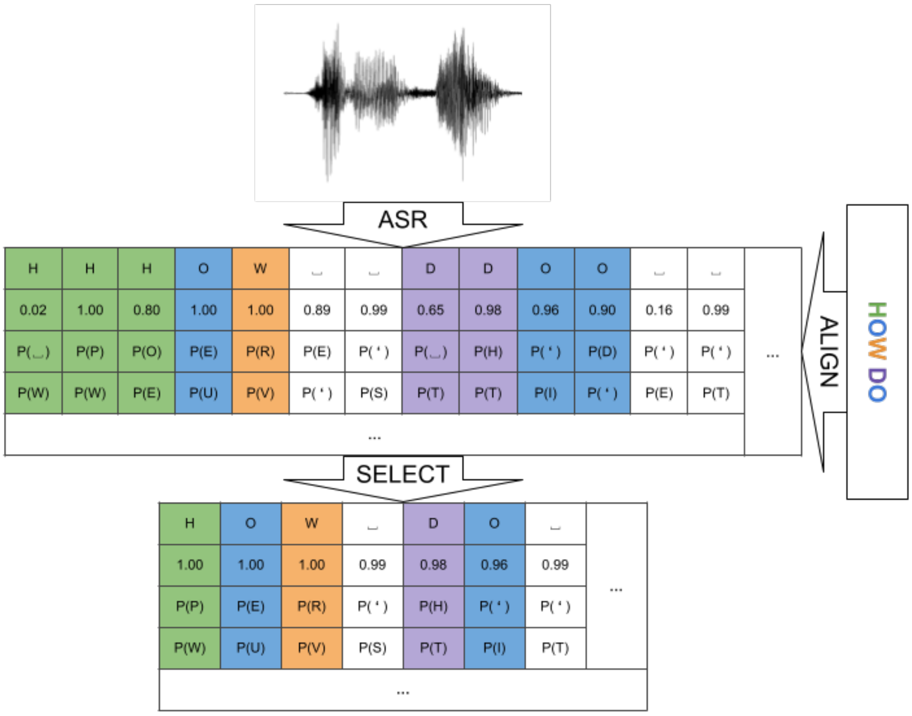
[2] https://www.memrise.com

[3] https://www.duolingo.com

[4] https://elsaspeak.com

[5] There is no meta data available about the individual language skills of those upvoting and downvoting.

[6] Double letters, such as the T's in letter, are handled by a special character prediction.



H	H	H	O	W	␣	␣	D	D	O	O	␣	␣	...
0.02	1.00	0.80	1.00	1.00	0.89	0.99	0.65	0.98	0.96	0.90	0.16	0.99	
P(␣)	P(P)	P(O)	P(E)	P(R)	P(E)	P(')	P(␣)	P(H)	P(')	P(D)	P(')	P(')	
P(W)	P(W)	P(E)	P(U)	P(V)	P(')	P(S)	P(T)	P(T)	P(I)	P(')	P(E)	P(T)	
...													

H	O	W	␣	D	O	␣	...
1.00	1.00	1.00	0.99	0.98	0.96	0.99	
P(P)	P(E)	P(R)	P(')	P(H)	P(')	P(')	
P(W)	P(U)	P(V)	P(S)	P(T)	P(I)	P(T)	
...							

Figure 1: The extraction process for retrieving one probability distribution per character from the audio clip. 1) Extract probability distributions for time slices using via ASR. 2) Align these to the elicitation phrase. 3) Then select one representative distribution per character. The first row of each table represents the highest probability character, the second row that character's probability, and the 3rd and 4th rows are the next highest probability characters. Each column contains a probability for each character, remaining character probabilities are represented by ellipses.

audio, represented by an arbitrary waveform, to the middle table. Each column in this table represents one time slice where the first row is the highest probability character, the second row is that character's probability (rounded to 2 decimal points), and the remaining rows indicate probabilities for other likely characters for this time slice. The model also predicts word boundaries, represented by a space (white columns). The next step aligns the probability distributions to the elicitation phrase, using a modification of the Needleman-Wunsch algorithm (see Section 5.2). The alignment is shown via the colors, e.g., all green columns align with the first character in the elicitation phrase. Based on this alignment, the best distribution (i.e., column) per character is chosen to represent the corresponding character in the elicitation phrase. The chosen distributions for each character are shown in the lower table in Figure 1.

Once we have an alignment between the probability distributions and true character labels, we need to choose one distribution per character in the elicitation phrase (i.e., one column per color, as shown in the lower table in Figure 1) to compare between speakers. This guarantees every character in the elicitation phrase is aligned to at least one probability distribution, even if the most probable character is not the true character. We decide which distribution, from all aligned candidates, to use for each character by choosing the single distribution where the probability of the true character is highest. These final distributions, one per true character, are what we compare between speakers to generate a score for each character.

The process to this point is executed on the learner and expert's pronunciations of the same phrase, resulting in two probability distributions per character of the phrase which we can compare pairwise. Since similarity comparisons are dependent on the similarity metric, we use three different algorithms for this comparison: cosine similarity, Jensen-Shannon Divergence (Lin, 1991), and Cross Entropy (see Section 5.3).

Elicitation phrase		H	O	…
Best hypothesis: expert		H	O	…
Best hypothesis: learner		H	O	…
Comparison	%	0.992	0.975	…
	Hel	0.034	0.097	…
	JSD	0.001	0.011	…
	XEn	0.016	0.047	…

Table 1: Example comparing the expert and learner and probability distributions (for the first two characters shown in Figure 1), resulting in a single score per character and similarity metric.

The pairwise comparison of the two speakers' productions per character is shown in Table 1. The probability distributions for each character per speaker is scored using the comparison algorithms, creating a single score per algorithm, which serves as feedback for each character. Since we do not know which similarity metric is the most suitable one, we experiment with three different ones (see section 5.3 for details).

4.2 Quantitatively Evaluating the Corpus

As discussed above, our goal is to evaluate the potential of Common Voice's annotation as a stand in for pronunciation annotation. I.e., we use the downvotes as indication for incorrect pronunciation. We use the vote annotations as our silver standard; the task then is to predict whether a given clip has any downvotes (irrespective of the number of upvotes) using ASR generated pronunciation scores. Assuming the pronunciation scoring algorithms work well, a classifier should be able to identify clips with downvotes. Since the number of votes per clip is small, we use a binary classification problem rather than predicting the number of downvotes. Most clips have a maximum of 3 total votes, and have 1 downvote and 2 upvotes if there are any downvotes. All clips have at least one upvote.

4.3 Data Preprocessing

We choose to focus on sets of files which contain at least 10 different speakers producing the same sentence. We then randomly sample 1 000 of these sets, containing 34 105 total utterances. Of these, the Coqui model fails to process 9,061 clips because of problems identified in preprocessing (e.g. the transcript contains unknown characters, or the clip is longer than 10 seconds). Our final count for clips is 25 044. Table 2 shows the Word Error Rate (WER) and Character Error Rate (CER) of the sampled data, along with the scores reported by Coqui for the used model when testing on the full dataset (in the version of 2021) (Coqui, 2021).

Dataset	WER	CER
Sampled Common Voice	0.252	0.153
LibriSpeech clean	0.052	0.019
LibriSpeech other	0.150	0.073

Table 2: Word Error Rate (WER) and Character Error Rate (CER) of sampled data used in our evaluation and Coqui AI's reported scores for English (Coqui, 2021).

By comparing the Coqui STT output of each clip with all other clips of the same sentence (see Section 5.3), we generate 511 532 comparisons. Since we define an expert utterance as one without downvotes, we only accept comparison pairs where one clip only has upvotes (expert) and the other as the language learner. To reduce the data to a manageable size given our compute resources, we reduce these randomly to 20 000 comparisons, split into 15 000 for training and 5 000 for testing.

5 System Components

5.1 Speech Recognition

We use the freely available model, Coqui STT[7] (Coqui, 2021), based on Baidu's DeepSpeech (Hannun et al., 2014). Out of the box, Coqui STT predicts an orthographic transcription of speech in an audio file by slicing it into chunks of a specified length (default: 20ms), and using an LSTM network to produce a softmaxed probability distribution over candidate characters per slice. This is illustrated in Figure 1 where the waveform is sliced into 20ms chunks, represented by the columns in the middle table. The rows represent probabilities of candidate characters.

Coqui STT was trained on approximately 47 000 hours of audio data from Common Voice (Ardila et al., 2020), LibriSpeech (Panayotov et al., 2015), and Multilingual LibriSpeech (Pratap et al., 2020). Both Librispeech corpora are comprised of segmented audiobook data.

Coqui STT's predictions over the sliced audio results in far more characters than the transcription; it decodes this long form transcription into the final predicted words using a Connectionist Temporal Classification (CTC) decoder (Graves

[7] `https://coqui.ai` (no longer maintained).

et al., 2006). We modify Coqui STT to preserve and return the softmax output in the form of probability distributions per 20ms time slice of the LSTM in the model's results, where the probability space is the set of all potential orthographic characters, thus bypassing the CTC decoder.

5.2 Needleman-Wunsch Alignment

Since we need to align the transcripts of the time slices to the correct transcription, rather than decoding the speech signal, we modify the alignment algorithm by Needleman and Wunsch (1970).

The algorithm's original purpose is to align two DNA sequences by calculating the distance between all possible alignments, using Levenshtein distance, and adding insertions to one or both sequences as needed. It then uses a backtrace to find the sequence resulting in the lowest divergence.

The original algorithm results in a $1 : 1$ alignment, with some characters aligned to an insertion character. When there are multiple possible alignments of equal weight, Needleman-Wunsch only returns the best entirely aligned sequences. However, for our problem, we need a many to one alignment, allowing us to be intentional about selecting a distribution per elicitation phrase character, rather than relying on the $1 : 1$ mappings. We modify the algorithm to allow pairing multiple items from the longer sequence (audio slices) with an item from the shorter sequence (correct transcription).

5.3 Comparing Distributions

We use three algorithms designed to compare probability distributions. The first is Hellinger Distance (Hellinger, 1909). It is a simple summation of comparisons between elements in the probability space normalized to be bounded by 0 and 1. The second is Jensen-Shannon divergence (JS; Lin, 1991). JS divergence is based on KL divergence (Kullback and Leibler, 1951), but it is symmetrical, making it a more consistent measure of similarity. It is also bounded by 1 when using probability distributions given the base of the log used is 2. The third is cross entropy. This is our only comparison metric which is not bounded by 0 to 1, and, like Jensen-Shannon divergence, a higher score indicates more dissimilar distributions.

Comparison Algorithm	Accuracy
Baseline	**81.4**
Jensen-Shannon	60.6
Cross Entropy	60.9
Hellinger	64.2

Table 3: Results per comparison algorithm scores as input to the downvote detection model.

5.4 The Downvote Detection Model

We evaluate our approach on the downvote detection task, trained on the comparison scores (see above). The downvote detection classifier consists of a Multi-Layer Perceptron with a softmax output layer, implemented using scikit-learn (Pedregosa et al., 2011). The goal of this classifier is a binary classification of whether a given clip has downvotes (indicating mispronunciation). The input features are the per character pronunciation scores from the distribution comparisons for each phrase. Phrases are of variable length, so the input is padded with ones to the length of the longest phrase. The final parameters are shown in Table 8 in the appendix. We optimized over the parameters using the Adam optimizer. The initial learning rate and beta 1 for Adam were the most impactful. More hidden layers did not improve performance, indicating that a complex network is not necessary for this task.

6 Quantitative Evaluation

In Table 3, we compare the accuracy of the downvote detection model when using the different comparison algorithms. The best results, 81.4%, are obtained by the baseline algorithm, using the probability of each character in the elicitation phrase from the speech recognition model's softmax. This is a binary classification with a $50 : 50$ split, i.e., random chance should yield about 50% accuracy. As an upper bound, 81.4% is therefore too low to be reliable. All of our comparison algorithm scorers perform at around 60-64%. They are similar to each other, with the Hellinger algorithm performing best after the baseline. This suggests that elaborate methods are not necessary for producing effective scores of pronunciation.

7 Qualitative Analysis

In this section, we probe deeper into the model, the task, and the corpus. If the vote annotation on

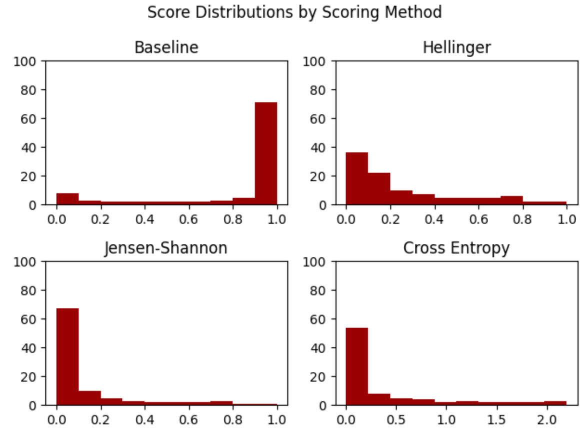


Figure 2: The distribution of pronunciation scores generated by the distribution comparison algorithms as percentages. The x-axis for Cross Entropy is different because it is not bounded by 0-1.

the clips in Common Voice are a reliable indicator of pronunciation quality, that should be reflected in the data. To test this, we choose a subset of instances we consider representative of the broader corpus with regard to both ASR performance and the mix of upvotes and downvotes.

7.1 Data in Aggregate

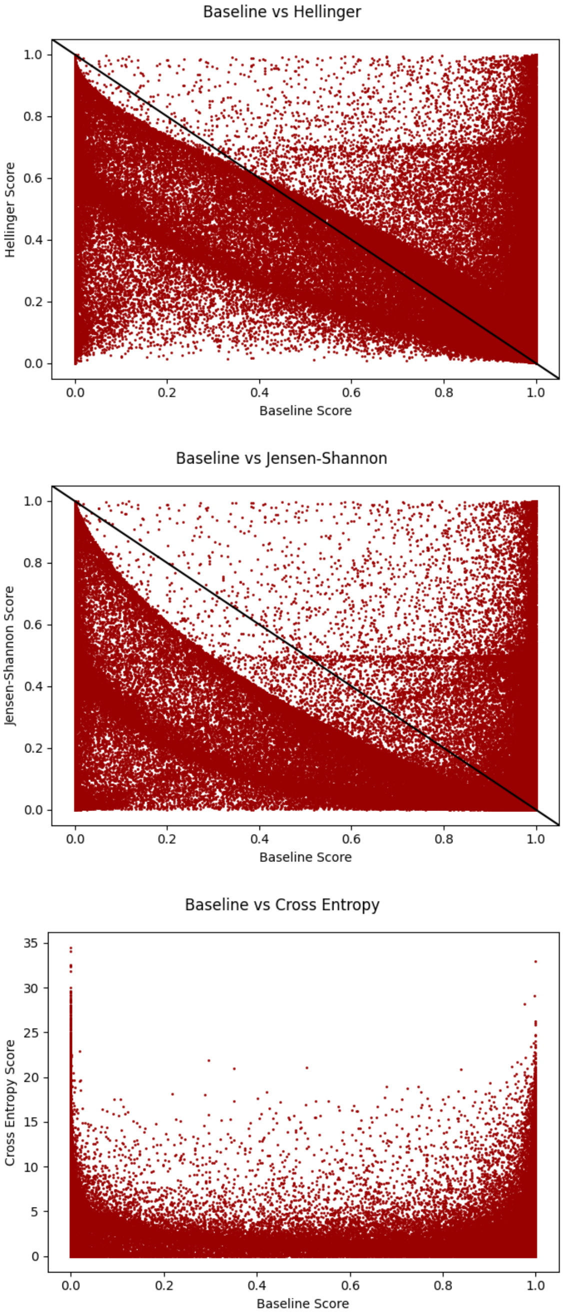
Figure 2 shows the distribution of scores by percent for each of the scoring methods. Each bin contains the output of the distribution comparison algorithm interpreted as a pronunciation score, within the bin's width of 0.1. Since cross entropy is not bounded by 1, its scores range to 35 for our data. However, such high scores are highly infrequent, thus we do not show scores >2. The scores generated from instances both with and without downvotes are included in these histograms. Separating the instances by presence of downvote results in nearly identical graphs.

For the baseline algorithm, the majority of scores are in the 0.9-1.0 bin. Since these are the probabilities given by the baseline for the character in the elicited phrase, this indicates that the ASR model is confident and accurate most of the time. This is expected for an English model, especially given the quantity of training data this model was trained on. The baseline model rarely returns intermediate probabilities. Consequently, when it predicts the wrong character or chooses no prediction, it still tends to do so confidently. The Jensen-Shannon scorer presents a similar pattern, the majority of scores are in the bin representing the best scores. (Since it is a distance metric, 0 represents the highest similarity and therefore a positive pronunciation score.)

The Hellinger scorer differs from the baseline, Jensen-Shannon, and Cross Entropy scorers in that it produces far fewer scores at the extremes of 0 and 1 or greater, instead making more distributed judgements. These differences indicate that some additional information is captured by the Hellinger scorer with regard to the relationship between the baseline and expert productions of the elicited phrase. The baseline scorer outperforming the Hellinger scorer (see Table 3) in our implicit evaluation task indicates that this relationship is not productive in predicting downvotes.

While the distributions in Figure 2 show an overview of the scorers, they do not directly compare the scorers to one another. We are most interested in how the comparison scorers relate to the baseline, as the baseline is representative of the model's confidence in its transcription. Figure 3 provides a direct comparison of the baseline scorer with the 3 scorers per character in each elicitation phrase. The diagonals provides a point of reference; scores above the diagonal are scored as worse pronunciation by the respective scorer for the same character, and scores below the diagonal are scored as better.

For the comparisons with the Hellinger distance and Jensen-Shannon divergence (top and middle of Figure 3), 1 on the y axis indicates a correct pronunciation, so the diagonal indicating agreement between the comparison and baseline has a negative slope. Most of the points appear below the agreement diagonal, showing that the scorers are more forgiving overall of mispronunciation. On both extremes of the x axis, 0 and 1, there is a broad range of scores on the y axis. As discussed above, this is where the majority of baseline scores appear, especially around 1, which is why the density at those extremes is much higher. From 0.9-1.0 on the x axis, the y axis has points ranging from 0-1, but the majority tend to be low, indicating that the Hellinger scorer tends to agree with the Baseline scorer when the ASR model is confident. There is more disagreement between the scorers at the 0 x axis extreme. This may be influenced by the smaller sample size compared to the 1 extreme, but there are enough points to confirm that the Hellinger scorer is more forgiving when the ASR model has low confidence. Of the non-


Baseline vs Hellinger

Baseline vs Jensen-Shannon

Baseline vs Cross Entropy

Figure 3: The relationship between scores in the baseline and the Hellinger, Jensen-Shannon, and Cross Entropy scorers. Each point represents a character's pronunciation score with the baseline on the x axis and the graph's respective comparison scorer on the y axis. Scores on the diagonal are equally scored by the baseline and the comparison scorer.

Baseline scorers, the Hellinger scorer performed best, which is likely due to the higher agreement it has with the Baseline.

The middle plot in Figure 3 compares the baseline with the Jensen-Shannon Divergence scorer. There is far less agreement in the Jensen-Shannon scorer than the Hellinger/Baseline comparison in the intermediate scores, but overall the Jensen-Shannon and Baseline scorers compare very similarly, being generally more forgiving when the ASR model has low confidence in its predictions.

Cross Entropy, unlike Hellinger Distance and Jensen-Shannon, is not bounded by 0-1, so there is no agreement diagonal in the bottom plot in Figure 3. Similar to Jensen-Shannon and Hellinger, most of the points are concentrated around the 0 and 1 extremes of the x axis. Because the Cross Entropy scores are on a much larger scale, creating a threshold for a mispronunciation would be at a different value than for the other scorers, and difficult to determine.

7.2 Specific Examples

As discussed in Section 3, the dataset used for these experiments is intended and annotated specifically for speech recognition, not for any specific pronunciation or dialect. This is, however, the closest available annotation to our task. The annotations on the audio clips collected indicate whether the speaker in a clip "accurately [spoke] the sentence", represented as upvotes or downvotes. Downvotes can indicate a mispronunciation, but also frequently indicate bad audio quality or missing audio. Conversely, upvotes do not distinguish between dialects, since a desired characteristic of ASR is the ability to generalize over dialect. We investigate a small number of examples further, relying on the first author's native American English judgments. In addition to looking into different issues resulting from the data, we are also interested in the question whether the different similarity metrics we used can provide complementary information to the baseline scores.

We take a closer look at individual examples from Common Voice, illustrating a range of issues, see Tables 4, 5, 6, and 7. Scores that show a distance > 0.3 from a perfect pronunciation (0 or 1, depending on the metric) are highlighted in red, indicating a mispronunciation.

Table 4 demonstrates the expected behavior in the case of a mispronunciation. The last word, `feel`, is mispronounced by learner 174840. The `f` is dropped and the `e`'s are pronounced as a lax high front vowels instead of tense. The ASR model is able to correctly transcribe the clip as `how do you feel`, though it reports being nearly equally confident that the last word is *hear*.

	h	o	w	d	o	y	o	u	f	e	e	l
Expert	0.995	0.801	0.962	0.965	0.882	0.918	0.880	0.901	1.000	1.000	0.998	0.999
Baseline	0.992	0.975	0.985	0.919	0.897	0.943	0.973	0.971	0.352	0.752	0.381	0.040
Hellinger	0.031	0.247	0.095	0.083	0.104	0.046	0.148	0.065	0.634	0.132	0.311	0.871
JSD	0.001	0.076	0.012	0.009	0.015	0.003	0.030	0.006	0.439	0.022	0.130	0.848
Cross Entropy	0.047	1.490	0.333	0.256	0.766	0.220	0.707	0.250	1.509	0.413	1.410	4.639

Table 4: Comparing Expert 167006 and Learner 174840.

	h	o	w	d	o	y	o	u	f	e	e	l
Expert	0.999	0.999	0.999	0.981	0.961	0.994	0.994	0.990	0.998	0.999	0.998	0.999
Baseline	0.992	0.975	0.985	0.919	0.897	0.943	0.973	0.971	0.352	0.752	0.381	0.040
Hellinger	0.034	0.097	0.060	0.165	0.119	0.024	0.055	0.034	0.629	0.132	0.309	0.872
JSD	0.001	0.011	0.004	0.033	0.028	0.001	0.004	0.002	0.437	0.022	0.130	0.848
Cross Entropy	0.016	0.047	0.026	0.186	0.347	0.100	0.066	0.118	1.514	0.415	1.406	4.640

Table 5: Comparing Expert 156711 and Learner 174840.

	h	o	w	d	o	y	o	u	f	e	e	l
Expert	0.999	0.999	0.999	0.981	0.961	0.994	0.994	0.990	0.998	0.999	0.998	0.999
Baseline	0.869	0.876	0.643	0.758	0.033	0.484	0.537	0.352	0.871	0.941	0.919	0.859
Hellinger	0.077	0.233	0.349	0.253	0.744	0.464	0.470	0.369	0.234	0.163	0.187	0.076
JSD	0.008	0.061	0.136	0.078	0.645	0.244	0.258	0.177	0.060	0.029	0.038	0.007
Cross Entropy	0.208	0.197	0.640	0.457	5.009	1.051	0.903	1.543	0.208	0.091	0.134	0.224

Table 6: Comparing Expert 156711 and Learner 103321.

While the Hellinger and Jensen-Shannon scorers capture these issues just as the baseline scorer does, the Cross Entropy scorer is much more critical, indicating errors where there are none in the first three words.

Table 5 shows the same learner as in Table 4, but compared with a different expert. The baseline scores are identical to Table 4 because they are independent of the expert. Though the expert scores are high in both Tables 4 and 5, the scores generated by the comparison scorers correctly indicate better pronunciation of the vowels in the first three words, especially in the Cross Entropy comparison. This demonstrates the impact that the selection of the expert has on scoring when using the comparison metrics, especially for the Cross Entropy scores. In the implicit evaluation, the comparison metrics perform worse than the baseline, but the impact of the choice of expert shows that there is at least some potential in those scorers which is not captured by that evaluation.

Table 6 contains an example where the expert speaker speaks clearly and the learner, though sounding native, does not enunciate clearly, so that the ASR model misunderstands `you` in the production, shown by the low scores. In this example, the forgiveness of the Jensen-Shannon scorer captures better that the learner pronounces the phrase correctly despite their lack of enunciation. The Hellinger scorer and cross entropy scorer closely reflect the baseline. This again shows the potential of the comparison scorers not captured by the implicit evaluation.

In Table 7, the expert speaker pronounces the phrase correctly, but the quality of the audio is very poor, and the ASR model has trouble transcribing the clip, though it is understandable to a native speaker. The learner also pronounces the clip correctly, i.e., the baseline scorer is correct in its feedback. The other scorers, however, incorrectly indicate mispronunciations in the learner's pronunciation. This is an issue with our expert selection more than with the annotation. However, in the case of this speaker being selected as a learner instead of a speaker, several characters would still be incorrectly marked as mispronounced. Choosing an expert carefully is critical, and in this case, the Common Voice annotation is not reliable enough to do so. Overall, we reveal an issue in our methodology for selecting the expert side of the comparison, specifically that the lack of any downvotes is a poor selection criterion, as poor quality clips may get through the annotation without any downvotes.

	h	o	w	d	o	y	o	u	f	e	e	l
Expert	0.271	0.074	0.000	0.770	0.903	0.973	0.988	0.979	0.997	0.952	0.001	0.000
Baseline	0.992	0.990	0.984	0.984	0.992	0.999	0.999	0.997	0.997	0.995	0.993	0.998
Hellinger	0.434	0.804	0.750	0.283	0.125	0.050	0.047	0.028	0.020	0.109	0.680	0.704
JSD	0.235	0.770	0.572	0.097	0.020	0.003	0.003	0.001	0.001	0.015	0.496	0.499
Cross Entropy	2.675	8.502	2.385	2.218	0.578	0.114	0.095	0.070	0.041	0.384	0.086	0.004

Table 7: Comparing Expert 18456694 (bad quality audio) and Learner 18400454.

8 Conclusion & Future Work

Our investigation has shown that the upvote and downvote annotations make a poor substitute for a properly annotated pronunciation corpus. Clips which have native sounding speech also have downvotes because of the poor audio. There is a great deal of variation in dialect and audio quality, which is desirable for training a speech recognition model, but represents noise when grading pronunciation. A downvote is far more commonly used as an indicator of an issue with the file itself than of a mispronunciation. The issue goes both ways as well, many clips with very poor audio quality have no downvotes but are not accurately processed by the speech recognition system. Most clips also have very few votes overall (most commonly 3), which prevents us from using ratios of up- and downvotes.

Many of the issues we identified, especially in section 7, indicate that there is a need for a speech corpus annotated for pronunciation. Many of the problems, such as selection of experts and variation in dialect and audio quality, can only be addressed by a careful collection of data and having clearly defined annotations.

As demonstrated in section 7.2, the comparison scorers still demonstrate some promise. Since the data situation makes it impossible to evaluate our scorers accurately, our next step is to collect a speech corpus annotated for pronunciation. We can then evaluate and continue to develop these scorers.

9 Limitations

We recognize that we make several critical assumptions throughout this work necessary to interpret our results: 1) Moses et al. (2020) show that using the ASR softmax probabilities per character is a reasonable way to score goodness of pronunciation. Our results indicate that either our model (see section 4.2) does not capture the relationship between pronunciation and downvotes, or there is none (the latter possibility being supported by our qualitative analysis). 2) There are no pronunciation corpora available with the type of annotations required for the task. In the absence of such data, we use the closest alternative. While it is possible to create such corpora for e.g. English, it may not be possible for many under-resourced languages. For the latter, using Common Voice may still be the only option. 3) We assume that the comparison metrics used are reliable. However, this can only be tested empirically once we have usable data. Finally, the ASR model is trained solely for speech recognition and not finetuned for the task of pronunciation. As we have no character level annotation to work with, finetuning is not possible in this work.

References

Chesta Agarwal and Pinaki Chakraborty. 2019. A review of tools and techniques for computer aided pronunciation training (CAPT) in English. *Education and Information Technologies*, 24:3731–3743.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Harry Bratt, Leonardo Neumeyer, Elizabeth Shriberg, and Horacio Franco. 1998. Collection and detailed transcription of a speech database for development of language learning technologies. In *ICSLP*.

Coqui. 2021. English stt v1.0.0. Technical Report STT-EN-1.0.0, Coqui, https://coqui.ai/models.

Sylvain Coulange. 2023. Computer-aided pronunciation training in 2022: When pedagogy struggles to catch up. In *Proceedings of the 7th International Conference on English Pronunciation: Issues and Practices*, pages 11–22.

Jonathan Dalby and Diane Kewley-Port. 1999. Explicit pronunciation training using automatic speech recognition technology. *CALICO Journal*, 16(3):425–445.

Horacio Franco, Victor Abrash, Kristin Precoda, Harry Bratt, Ramana Rao, John Butzberger, Romain Rossier, and Federico Cesari. 2000. The SRI EduSpeak(TM) system: Recognition and pronunciation scoring for language learning. In *Proceedings of InSTILL*, pages 123–128.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Ernst Hellinger. 1909. Neue Begründung der Theorie quadratischer Formen von unendlich vielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 136:210–271.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. 2021. Automatic speech recognition: A survey. *Multimedia Tools and Applications*, 80(6):9411–9457.

Caleb Moses, Miles Thompson, Keoni Mahelona, and Peter-Lucas Jones. 2020. Scoring pronunciation accuracy via close introspection of a speech recognition recurrent neural network. In *NeurIPS2020*.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Ambra Neri, Catia Cucchiarini, and Helmer Strik. 2006. ASR-based corrective feedback on pronunciation: Does it really work? In *Proceedings of Interspeech*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Silke M Witt. 2012. Automatic error detection in pronunciation training: Where we are and where we need to go. In *International Symposium on Automatic Detection on Errors in Pronunciation Training*.

Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018. L2-ARCTIC: A non-native English speech corpus. In *Proceedings of Interspeech*, page 2783–2787.

A Model Parameters

Best Model Parameters	
input embedding	152
hidden layer size	128, 64, and 32
activation	ReLU
optimizer	Adam
batch size	200
learning rate	5e-4
Adam beta 1	0.80

Table 8: Optimized model parameters for the implicit evaluation.