Comparing Phonological Feature Sets for Low-Resource ASR*

Alessio Tosolini[1], Massimo Daul[2], Ayla Karakaş[3], & Claire Bowern[3]
[1]*McGill,* [2]*NYU,* [3]*Yale University*

1 Introduction

Automatic Speech Recognition (ASR) tools have proven to be helpful to linguists in a variety of language documentation tasks, including reducing the amount of time spent transcribing speech audio completely from scratch. (Prud'hommeaux et al., 2021; Cavar et al., 2016).

The task of an ASR model that can phonetically transcribe speech is to learn a direct mapping from utterance waveform to token sequence (phones or orthographic symbols). This design is poorly matched to low-resource settings: when training data is scarce or low-quality, models must infer language-specific distinctions from limited evidence, and the resulting systems are difficult to adapt to new languages because their output vocabulary is fixed.

Training ASR models to encode more phonetically informed structure, such as articulatory features, has yielded positive results for transcription in a variety of deep learning architectures (Koreman et al., 1998; Xu et al., 2021; Taguchi et al., 2023; Lee et al., 2025). (See Jimerson et al. (2023) for discussion of the interplay between model and language more generally.)

In this paper, we explore an alternative ASR framework in which phonological features are predicted as an explicit intermediate representation, rather than predicting phones directly. Because feature systems encode cross-linguistically meaningful structure, this intermediate representation can reduce sample complexity by constraining what must be learned from limited data, while also enabling rapid adaptation to new languages through changes to the phone-to-feature mapping rather than retraining the model. As a result, this approach is particularly well suited to low-resource settings.

To test our approach, we experiment with `Phonet` (Vásquez-Correa et al., 2019), a bank of parallel bidirectional RNNs that extract phonological features from speech audio by estimating their posterior probabilities. Despite generally being outperformed by modern ASR systems, Phonet models remain valuable for a relatively straightforward comparison between featurally informed ASR and phoneme level ASR.

In this proof-of-concept work, we retrained Phonet models on two different feature sets to see the extent to which specific theories of phonological features facilitate better phoneme recognition, using a low-resourced language (Yan-nhangu, Pama-Nyungan) as a testing ground for performance. We use a naïve greedy decoding strategy to isolate the effect of feature set choice, and find that IPA features lead to the best transcription accuracy, followed closely by a featureless baseline.

2 Methods

2.1 *Phonet architecture* Phonet (Vásquez-Correa et al., 2019) is an RNN originally made to detect 'phonological posteriors', groups of features that carry information about the speech of a speaker. These phonological posteriors are functionally equivalent to activations of phonological features, where for any given speech frame, a value from 0 to 1 is assigned for every phonological feature specified prior to model

* Work division: Developed the project: MD, AT, AK; Sourced and processed data: MD, AT, CB; trained models: MD, AT; analyzed results: MD, AT, AK, CB; wrote paper: MD & AT with input from CB

Proceedings of *AMP 2025*

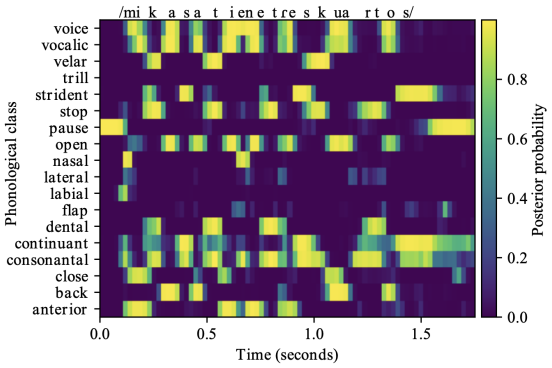

Figure 1: Posteriorgram with phoneme activations from a Spanish speech sample from Vásquez-Correa et al. (2019).

training. Phonet has been used in many settings that require quantifying the variability in the extent to which a speech sound exhibits a certain feature, such as the height of Italian mid-vowels (Jones and Renwick, 2024) or consonant lenition as a marker for Parkinsons Disease (Wayland et al., 2025). Given multiple features, Phonet outputs a phonological activation for each of the features specified for each of the frames in the input audio, which we call a posteriorgram. A sample posteriorgram taken from (Vásquez-Correa et al., 2019) from a Spanish sample is seen in Figure 1. Finally, by specifying phonemes instead of features, Phonet can output activations for each phoneme per frame, giving us a posteriorgram of phoneme activations instead of feature activations.

Phonet models are trained on phone-level annotated speech with a dictionary mapping phonological classes to a set of phones. For an overview of Phonet's architecture, see Figure 2.

2.2 *ASR Pipeline* For most ASR models, phonetic information is captured implicitly via a learned relationship between audio and a pre-defined orthography. Given an audio input, a traditional ASR model first converts the signal into a sequence of frame-level acoustic feature vectors. These vectors are then mapped—via a learned encoder–decoder or CTC-style architecture—to a sequence of token predictions aligned at the frame level. Finally, a collapsing or decoding step removes repetitions and blank symbols, producing a shorter token sequence corresponding to the predicted transcription. The comparison is schematized in Figure 3.

Let the acoustic input be represented as a sequence of frame-level feature vectors extracted from an input waveform,

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \qquad \mathbf{x}_i \in \mathbb{R}^d,$$

where n is the number of audio frames and d is the acoustic feature dimension.

A conventional ASR model learns a direct mapping

$$f_{\text{ASR}} : \mathbb{R}^{n \times d} \to \mathcal{V}^n,$$

where $\mathcal{V}$ is a discrete token vocabulary and predictions are made at the frame level. A collapsing function $C(\cdot)$ (e.g., CTC decoding) then produces a shorter output sequence (

$$\mathbf{Y} = C(f_{\text{ASR}}(\mathbf{X})), \qquad \mathbf{Y} = (y_1, \ldots, y_m), \quad m \leq n.$$

At a high level, this process can be summarized as

$$\mathbf{X}_n \longrightarrow \mathbf{Y}_m,$$

where phonetic structure is implicitly encoded within the learned audio-to-token mapping rather than modeled as an intermediate representation.

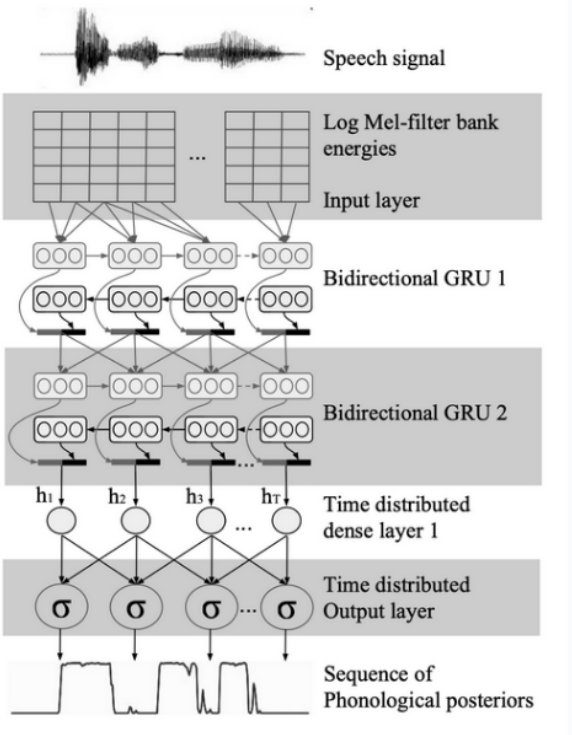

Figure 2: Architecture of the Phonet system showing the inference process for phonological posteriors from the speech signal

In this paper, we propose a different approach to ASR. As in traditional ASR models, we initially convert the input audio into a sequence of acoustic feature vectors. Then, instead of directly predicting discrete phonemic or orthographic tokens at each frame, we generate activations over a set of phonetic features. This yields a sequence of continuous feature-valued representations of the audio frames. We can then associate these feature activations to phone-level predictions. For each frame, the generated posteriorgram consists of real-valued activations corresponding to the presence of each phonetic feature. These activations are compared against the feature vectors for each phone in the target language's inventory, and the the closest phone is selected under a distance metric (Currently Euclidean). Finally, the resulting frame-level phone sequence is collapsed using the algorithm described in 2.2.1 to produce the final phoneme string.

Then, this proposed model learns a mapping from acoustic features to phonetic feature activations

$$f_{\text{feat}} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times k},$$

where k is the number of phonetic features and each output vector $\mathbf{p}_i \in [0,1]^k$ represents a frame-level phonetic feature posterior.

To get discrete phone predictions, each frame-level feature activation vector $\mathbf{p}_i \in \mathbb{R}^k$ is mapped to the closest phone in the phonetic inventory $\mathcal{P}$ by Euclidean distance in the feature space:

$$\hat{y}_i = \arg\min_{j \in \mathcal{P}} \|\mathbf{p}_i - \boldsymbol{\phi}_j\|_2,$$

where each $\boldsymbol{\phi}_j \in \{0,1\}^k$ denotes the feature vector from our phonological feature set associated with phone j. For our purposes, these are PHOIBLE or IPA feature vectors.

This yields a frame-level phone sequence $(\hat{y}_1, \ldots, \hat{y}_n)$, which is collapsed (see 2.2.1) to produce the final output

$$\mathbf{Y} = (y_1, \ldots, y_m), \qquad m \leq n.$$

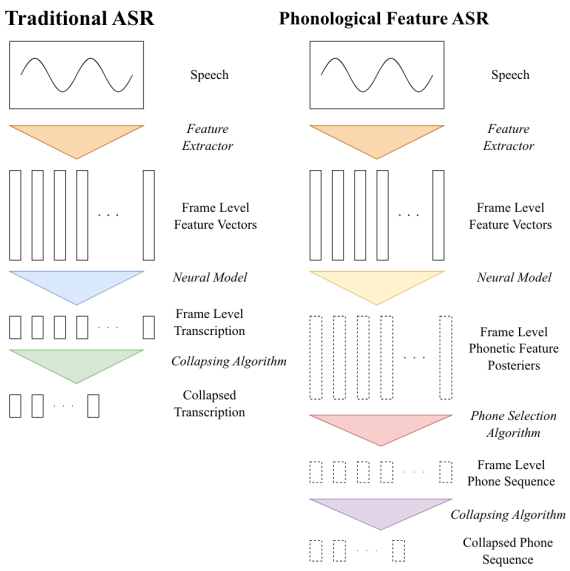

Figure 3: Comparison between traditional ASR and ASR which uses phonological features

At a high level, the proposed pipeline is summarized as

$$\mathbf{X}_n \longrightarrow \mathbf{P}_n \longrightarrow \mathbf{Y}_m,$$

where $\mathbf{P}_n$ denotes this intermediate sequence of phonetic feature representations.

2.2.1 *Collapsing Algorithm* Sounds are often longer than a single audio frame, which means that the predicted sequence of phones will contain many repeated values and possible poor predictions. To account for this, we collapse that sequence.

The simplest collapsing strategy removes consecutive duplicate phoneme labels while preserving their original order. The algorithm iterates through the frame-level sequence from left to right and appends a phoneme to the output only when it differs from the most recently retained symbol.

In addition we experimented with a local-frequency filter designed to remove weak / spurious phoneme predictions. For each frame, the predicted phoneme is retained only if it occurs at least r times within a symmetric window of size w centered on the current frame. The filtered sequence is then collapsed using the same duplicate-reducing procedure described above.

As an example, consider an audio clip containin the word ”feature”. Our model may produce a sequence of predicted phonemes such as:

[p, b, b, a, ə, ə, n, n, a, a, ə, n, n, ə, ə]

This would be frequency filtered to obtain

[b, b, ə, ə, n, n, a, a, n, n, ə, ə]

and then reduced to

[b, ə, n, a, n, ə]

This approach is simplistic and is evaluated further in the discussion.

IPA Features	PHOIBLE Features
voiced, bilabial, velar, palatal, dental, alveolar, retroflex, glottal, plosive, nasal, trill, lateral, approximant, vocalic, high, low, front, center, back, round, short, long, silence	syllabic, long, consonantal, sonorant, approximant, trill, nasal, lateral, labial, round, coronal, anterior, distributed, dorsal, high, low, front, back, tense, periodicGlottalSource, constrictedGlottis, silence

Table 1: Features used for the two featural models.

2.3 *Feature sets* Put simply, features are ways of exhaustively classifying phoneme elements using binary (present/absent) characteristics. They have been used in phonology since at least Chomsky and Halle (1968). Arguments vary regarding the extent to which features are abstract elements that delineate contrasts, or are more concretely grounded in acoustics and/or articulation.

The first feature set we investigate is a naïve feature set consisting of the place and manner of articulation, as seen on the left in Table 1. In it, we can find the seven places of articulation and five manners of articulation that Yan-nhangu consonants are specified for, on top of the binary feature for voice. For vowels, we include vocalic, as well as the two heights, three backness contrasts, and two lengths that Yan-nhangu vowels may take. We also include the feature silence, which is specified as 1 for any audio frame that has no annotation and 0 elsewhere.

PHOIBLE features are adapted from Moran and McCloy (2019), with features loosely based on the geometry in Hayes (2009) with additions from Moisik and Esling (2011). Features were automatically generated the entries for the each of the phonemes in Yan-nhangu from PHOIBLE, with features with the same value for all phonemes removed (e.g. click, which is specified as 0 for all phonemes in Yan-nhangu). We similarly add silence as a feature.

Finally, since Phonet is able to output phoneme posteriorgrams in the same way it's able to generate feature posteriorgrams, we trained a model to predict individual phonemes for a baseline.

2.4 *Language data* The corpus for this test includes selected field recordings of the Yan-nhangu language, a member of the Yolŋu subgroup of Pama-Nyungan (Glottolog code YANN1237; ISO-639 JAY). Documentary materials for Yan-nhangu were recorded in the period 2004–2007 at the request of elders of the Mäl̲arra, Gamalaŋga, and Gorryindi clans, all of whom have now passed away.

Yan-nhangu's phoneme inventory includes both place and manner contrasts, with stops at 6 points of articulation and an additional glottal stop. There are two stop series distinguished by both length and VOT, and 6 vowels distinguished by height, length, and frontness. Tables 2 and 3 give the consonant and vowel inventory of Yan-nhangu.

p	t	ʈ (t̲)	ṯ (th)	c (tj)	k	ʔ (')
b	d	ɖ (d̲)	ḏ (dh)	ɟ (dj)	g	
m	n	ɳ (n̲)	ṉ (nh)	ɲ (ny)	ŋ	
	l	ɭ (l̲)				
	r (rr)	ɻ (r)				
w				j (y)		

Table 2: Consonant inventory of Yan-nhangu

i	u	iː (e)	uː (o)
a		aː (ä)	

Table 3: Vowel inventory of Yan-nhangu

All phonet models were trained on 3.07 hours of Yan-nhangu phone-aligned input. 5 speakers were in

the recordings, which ranged in the number of speakers per recording. In total, there were a total of 4638 training audio snippets of average duration 2.4s. Three models were trained on this data, a featural model trained on IPA features, a featural model trained on PHOIBLE features, and a phone-level baseline against which we can compare the featural method against.

Testing was performed on three langauges: Yan-nhangu (19.2 mins), Yidiny (Pama-Nyungan; 39.0 mins), and Kunbarlang (Arnhem; 16.1 mins). We ran both featural models and the baseline phone-level model on each of the three testing languages and evaluated accuracy using the Phone Error Rate (PER) between the manual transcription and the ASR prediction. No fine-tuning was performed when evaluating the models on Yidiny or Kunbarlang, instead, a custom mapping from the IPA features and PHOIBLE features to the phonemes in Yidiny and Kunbarlang inventories was manually produced to show the ease with which featural ASR can be adapted to new languages in a zero-shot setting. No equivalent conversion can be made for the phone-level baseline, so the comparison between the PER of the phone-level baseline and the featural models for Yidiny and Kunbarlang is not directly interpretable.

2.5 *Models* We retrained three Phonet models: two models predicting features from a predefined feature set (IPA or PHOIBLE features) and one baseline model predicting a phone sequence directly from the audio (no features). Featural models output posteriorgrams, matrices storing an activation from 0 to 1 for each feature for every 10 ms audio frame. To decode, for each audio frame, we computed the Euclidean or cosine distance between the posteriorgram's feature activations to the feature vector of each phone in the target language's inventory. A phone was predicted greedily by selecting the phone that minimizes distance in the resulting distance vector over the target phones. Since multiple audio frames can span a single phone, our naïve decoding strategy requires that for a window length of n phones in either direction, a phone is removed if it occurs less than m times in that window (n=3, m=4 worked best). Collapsing consecutive identical phones after this removal yields a predicted phone sequence, which was used to calculate average phone error rate (PER) against a human annotated label.

2.6 *Adaptation to Other Languages* One of the major advantages in using features for ASR is the supposed universality of features across the world's languages. This paper adapts the featural models to Yidiny and Kunbarlang testing data by defining new phone to feature mappings for these languages for the phones in the inventories of these languages. No additional finetuning was performed for these languages. Note that due to differences in the inventories of Yan-nhangu to those of Yidiny and Kunbarlang, it was not possible to create a mapping between the output of the Yan-nhangu phone-level model and that of the Yidiny and Kunbarlang models. For this reason, it is not possible to create a direct comparison between the featural models and the phone-level baseline for Yidiny and Kunbarlang.

3 Results

This paper seeks to demonstrate the feasibility of a featural approach to ASR, in which phonetic information is explicitly modeled in an intermediary step, rather than learned implicitly as in traditional audio-to-orthography systems. To evaluate this approach, we train three models as described in § 2.5. The phone level baseline is a Phonet model trained to identify phonemes instead of features and is used as a baseline against which we can test the effectiveness of our alternative methodology. This provides an effective baseline for Yan-nhangu since the model architecture is the same as that for the featural models.

	Yan-nhangu (Training Language)	**Yidiny**	**Kunbarlang**
IPA features	0.42145	0.69762	0.73668
PHOIBLE features	0.52793	0.72947	0.80081
Phone-level (baseline)	0.45352	*0.79052*	*0.81166*

Table 4: Performance comparison across feature representations and languages.

Table 4 presents the PER for the three models across the two distance metrics. We found the model using IPA features yields the lowest PER of 42.1%. This model outperforms both the phone-level baseline and the PHOIBLE featural model, with the PHOIBLE featural model performing the worst out of the three models by a large margin. For Yidiny and Kunbarlang, the margin between the performance of the IPA featural

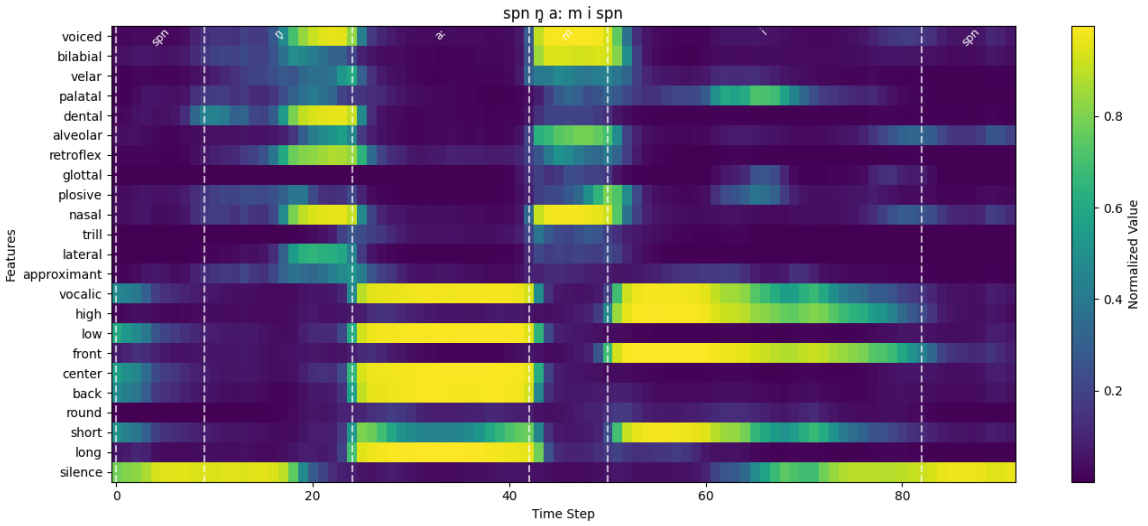

Figure 4: Posteriorgram for /n̪ami/ using the IPA feature set.

model and the PHOIBLE featural model was less large than for Yan-nhangu, but for both testing languages performance was best for the IPA featural model. Since phonetic inventory of Yan-nhangu is not a clear superset of that of Yidiny and Kunbarlang, the phone-level baseline was not able to be adapted to the two testing languages. As a result, no direct comparison can be made to the phone-level baseline for Yidiny and Kunbarlang.

3.1 *Yan-nhangu Testing Setting* For the Yan-nhangu testing setting, the best performing model was the IPA features model. Using this model, we see approximately a 3% improvement by using featural ASR with the set of IPA features as opposed to the phone-level baseline. The featural model using the PHOIBLE feature set performs worse than the IPA features model and the phone-level baseline.

Figure 4 shows a sample posteriorgram for the Yan-nhangu word /n̪ami/ using the IPA feature model, which was correctly predicted as such by both the IPA and PHOIBLE models. In the figure, brighter colors correspond to high feature activations for a given timeframe, while lower activations correspond to lower feature activations, while vertical lines correspond to the manually annotated phone boundaries for each phoneme. For example, the feature *nasal* has a very high activation during the articulation of the nasal consonants /n̪/ and /m/, but very low values during the articulation of the oral vowels /aː/ and /i/, with gradient values between their articulation.

This is additionally seen in a different light in Figure 5, which isolates the feature activations for *dental, alveolar, nasal*, and *high*. In this figure, we see high activations for the feature *nasal* during the articulation of the nasal consonants only. Similarly, we see a high activation for *dental* during the articulation of /n̪/, but no other consonant. Interestingly, we see a medium activation for the feature *alveolar* for both nasal consonants. This activation is indicative of a model that is not confident in the place of articulation of the first consonant, but that believes that the /n̪/ is most likely ot be dental as opposed to other possibilities.

Figure 6 shows a posteriorgram for the same Yan-nhangu word, but using the model for PHOIBLE features. In this posteriorgram, we see features with clear parallels to the IPA feature model showing similar activations. Namely, the features *nasal* and *high* activate for the nasal consonants and high vowel respectively. However, in the absence of a clear *dental* or *alveolar* feature, we see more varied activation. Dental consonants are represented as the combination of the features *coronal* and *distributed*, which we see here with the high activations for the two features.

3.2 *Yidiny and Kunbarlang Testing Settings* Compared to Yan-nhangu, the PER for Yidiny and Kunbarlang is much higher (see Table 4), with both testing languages performing better given the IPA

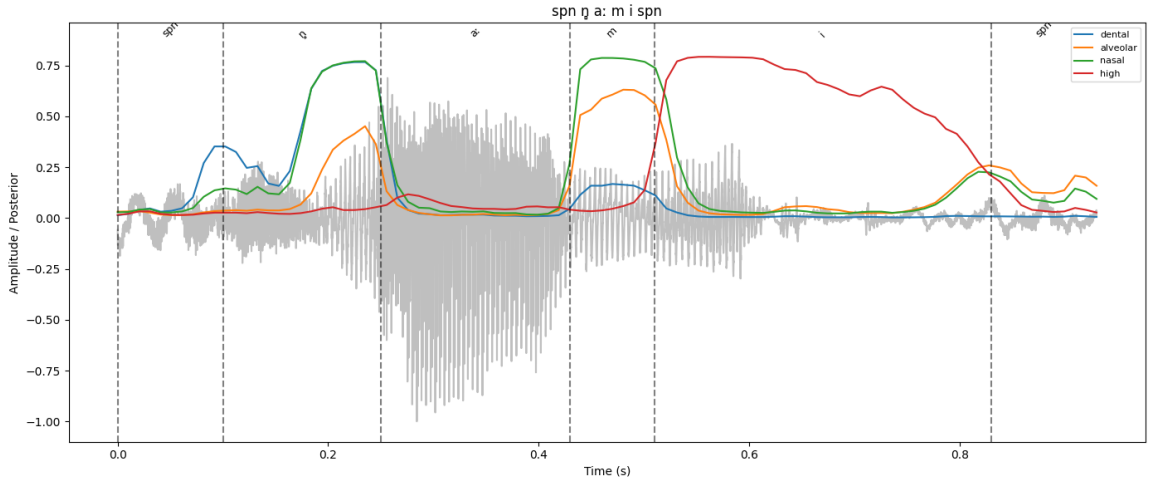

Figure 5: Posteriorgram and waveform for /n̟ami/ showing only the features dental, alveolar, nasal, and high taken from the IPA featural model.

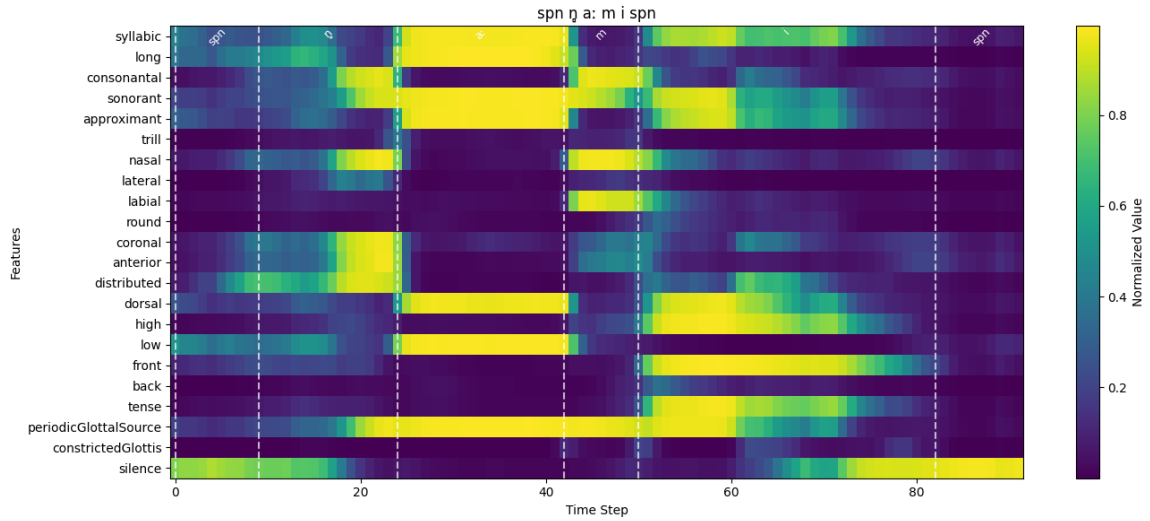

Figure 6: Posteriorgram for /n̟ami/ usign the PHOIBLE feature set.

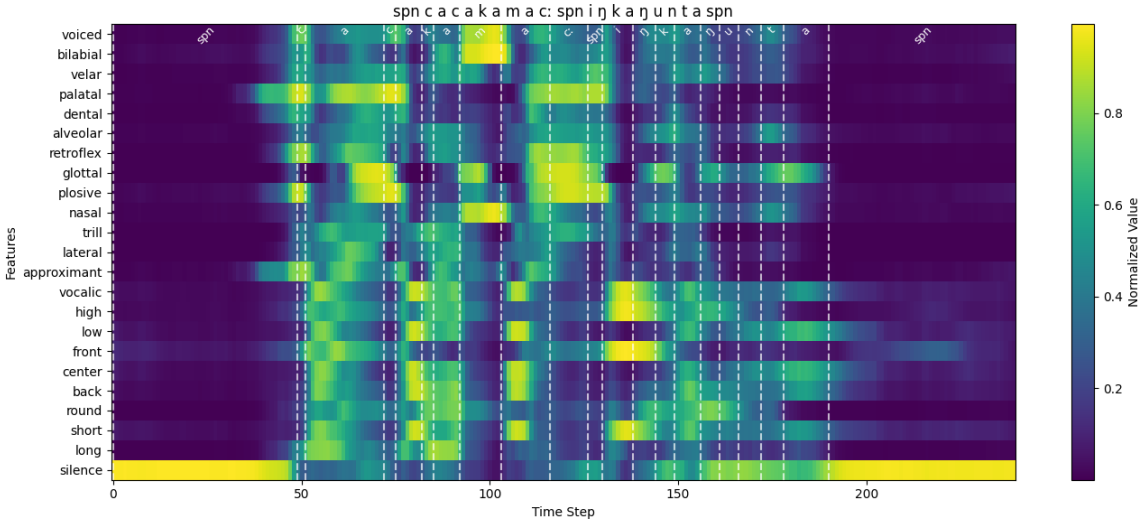

Figure 7: Posteriorgram for the Kunbarlang phrase /cacaka**m**ac: **i**ŋkaɲunta/ using the IPA feature set. The model predicted the output /naːʔcau**ː**ma**ʔ**c**i**u a/ (phonemes bolded for emphasis).

featural model compared to the PHOIBLE featural model. Looking at Figure 7 we can see the sample Kunbarlang phrase /cacakamacː iŋkaɲunta/ transcribed using the IPA model, where the model predicts the output /naːʔcauːmaʔciu a/. In it, we see that the activations for all features besides *silence* are messier compared to the Yan-nhangu data. The only clear feature actiations are for the phonemes /m/ and /i/, both of which are correctly predicted by the model and bolded in the caption for Figure 7 for emphasis. In these models, we see activations for features that do not appear in the original input audio, such as activations for the feature glottal, resulting in the model predicting the presence of /ʔ/ twice though it is absent in the input data. This activation may be indicative of the stops in Kunbarlang being glottalized more than in Yan-nhangu, suggesting that while features may be helpful for the purposes of ASR, they may also realize differently in different languages.

4 Discussion

Unlike canonical phone-level ASR models, which must learn language-specific token distinctions from limited data, featural models operate over a representation space grounded in known articulatory and phonological structure. This allows the model to leverage pre-existing relationships between phones and their featural descriptions, reducing the burden of learning in low-resource settings.

Across all three testing languages, we see lower phone error rate for the models trained on the IPA features compared to the PHOIBLE features. The reason for this is not entirely clear, as the PHOIBLE features were designed to more directly correspond to articulatory properties than the IPA features. While these two feature sets handle most features similarly, they differ in the handing of the non-peripheral stop series (e.g. /t, t̪, c, ʈ/). These four phonemes, along their nasal and voiced counterparts (for Yan-nhangu), are represented by only two features for the PHOIBLE featural model, with each phonemes being a unique combination of the features *distributed* and *anterior*. Compare this to the IPA model, where these stops are represented as having the feature *alveolar*, *dental*, *palatal*, or *retroflex*. This increase in the number of features would likely be more robust to noise in the data, leading to an overall PER rate. This may also explain the much smaller difference in IPA featural model vs PHOIBLE featural model performance for Yidiny, which lacks dental or retroflex stops, making the feature *anterior* sufficient to distinguish between the series. More research into which set of features is optimal needs to be conducted, especially in the context of zero-shot language transfer.

This pipeline serves as a proof of concept to demonstrate that it is possible to train models to learn

features instead of phonemes for improved ASR. However, there are some major limitations with the scope of this experiment.

While introducing an intermediate step may allow for improved performance and interpretability, it also means that errors will compound. As with many modular tools, even slightly inaccurate upstream steps may significantly reduce precision later on.

Second, the decoding and collapsing strategy is naive and cannot handle many cases, such as as a language with geminates. Borrowing methodology from CTC collapse, such as a blank-token or non-greedy search mechanism, is an important direction for future work.

Finally, the Phonet architecture is outdated and not designed for this use-case. Rebuilding a more sophisticated transformers-based model for predicting posteriorgrams will likely improve performance.

5 Conclusions

We presented preliminary results as a first step towards a more comprehensive exploration of the impact of different assumptions about the feature space in phonetically informed ASR models, compared to phoneme level ASR. We found that of our four configurations of feature set vs. distance metric, models making use of the place/manner of articulation features used for the IPA and a decoding strategy that uses Euclidean distance performed best. However, usage of more sophisticated decoding strategies, e.g., incorporating beam search, may reveal a different story. Future work aims to include such strategies, as well as expanding to cross-linguistic comparison, and qualitative analysis of errors in different feature systems.

References

Cavar, M., Ćavar, D., and Cruz, H. (2016). Endangered language documentation: Bootstrapping a Chatino speech corpus, forced aligner, ASR. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4004–4011.

Chomsky, N. A. and Halle, M. (1968). *The Sound Pattern of English*. Harper & Row.

Hayes, B. (2009). *Introductory Phonology*. Blackwell.

Jimerson, R., Liu, Z., and Prud'hommeaux, E. (2023). An (unhelpful) guide to selecting the best ASR architecture for your under-resourced language. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1008–1016, Toronto, Canada. Association for Computational Linguistics.

Jones, A. and Renwick, M. E. (2024). Evaluating italian vowel variation with the recurrent neural network phonet. In *Proc. Interspeech 2024*, pages 3679–3683.

Koreman, J., Andreeva, B., and Barry, W. J. (1998). Do phonetic features help to improve consonant identification in ASR? In *Proc. ICSLP 1998*, page paper 0549.

Lee, J., Mimura, M., and Kawahara, T. (2025). *Leveraging IPA and Articulatory Features as Effective Inductive Biases for Multilingual ASR Training*.

Moisik, S. R. and Esling, J. H. (2011). The 'Whole Larynx' Approach to Laryngeal Features. *International Congress of Phonetic Sciences*, 17:1406–1409.

Moran, S. and McCloy, D. (2019). PHOIBLE 2.0.

Prud'hommeaux, E., Jimerson, R., Hatcher, R., and Michelson, K. (2021). Automatic Speech Recognition for Supporting Endangered Language Documentation.

Taguchi, C., Sakai, Y., Haghani, P., and Chiang, D. (2023). Universal Automatic Phonetic Transcription into the International Phonetic Alphabet.

Vásquez-Correa, J., Klumpp, P., Orozco-Arroyave, J. R., and Nöth, E. (2019). Phonet: A Tool Based on Gated Recurrent Neural Networks to Extract Phonological Posteriors from Speech. In *Interspeech 2019*, pages 549–553. ISCA.

Wayland, R., Meyer, R., and Tang, K. (2025). Speech markers of parkinson's disease: Phonological features and acoustic measures. *Brain Sciences*, 15(11):1162.

Xu, Q., Baevski, A., and Auli, M. (2021). *Simple and Effective Zero-shot Cross-lingual Phoneme Recognition*.

6 Abstract

Automatic speech recognition (ASR) tools can substantially accelerate language documentation by reducing the effort required to produce time-aligned transcriptions from field recordings. However, most ASR systems are trained to predict a fixed inventory of discrete symbols (e.g., phones), making them difficult to adapt to low-resource settings and to new languages with different inventories. We investigate a featural ASR pipeline in which a model predicts phonological feature activations as an explicit intermediate representation, which is then decoded into phone sequences via phone–feature mappings.

Using `Phonet` (Vásquez-Correa et al., 2019), we train two featural models on approximately 3 hours of Yan-nhangu (Pama-Nyungan) and compare (i) an IPA place/manner feature set and (ii) a PHOIBLE-derived feature set (Moran and McCloy, 2019) against (iii) a phone-level baseline with the same architecture. On Yan-nhangu, the IPA-style featural model yields the lowest phone error rate, outperforming both the PHOIBLE featural model and the phone-level baseline. We perform zero-shot transfer to Yidiny (Pama-Nyungan) and Kunbarlang (Arnhem) by substituting new phone–feature mappings within the same model. For both languages, the IPA feature set outperforms PHOIBLE under the same decoding procedure. These results provide preliminary evidence that featural representations can be a useful intermediate step for low-resource phonetic ASR and may facilitate zero-shot transfer learning for languages with different phoneme inventories.