Spoken Document Retrieval for an Unwritten Language: A Case Study on Gormati

Anonymous ACL submission

Abstract

Speakers of unwritten languages have the potential to benefit from speech-based automatic information retrieval systems. This paper proposes a speech embedding technique that facilitates such a system that can be used in a zero-shot manner on the target language. After conducting development experiments on several written Indic languages, we evaluate our method on a corpus of Gormati – an unwritten language – that was previously collected in partnership with an agrarian Banjara community in Maharashtra State, India, specifically for the purposes of information retrieval. Our system achieves a Top 5 retrieval rate of 87.9% on this data, giving the hope that it may be useable by unwritten language speakers worldwide.

1 Introduction

Introducing and integrating well-designed digital systems into communities, particularly those with low digital participation, such as oral communities, can enhance their exposure to digital technologies and could reduce inequalities arising from their limited digital use or presence (Deumert, 2014; Gorman et al., 2011).

One application of advancements in language technology is in the application of speech-based search and information retrieval (IR). This task, commonly known as Spoken Document Retrieval (SDR) has been investigated over several decades, mostly notably in DARPA and IARPA programmes such as BOLT, GALE and Babel (Griffitt and Strassel, 2016; Olive et al., 2011; Hartmann et al., 2017) . Early work, e.g., (Weintraub, 1993) took the form of simple keyword-spotting tasks (sometimes referred to as Spoken Term Detection), but more sophisticated search capabilities have also been developed (Coden et al., 2002).

In a standard setting, SDR operates over spoken documents (i.e., audio and video files containing speech) but input queries remain text-based. However, in an alternative setting the input query may be in the form of speech as well. It is this latter formulation that is of course, most relevant to unwritten languages. Whilst SDR systems are typically developed for a specific target language, often using significant quantities of transcribed speech data for model training, this is not possible for an unwritten language. In this case, work to date has adopted a significant simplification of the IR task to that of Query-by-example (QbE), essentially a form of keyword-spotting in which spoken documents are ranked based on the estimated occurrence of an arbitrary spoken input phrase.

QbE systems have been developed in a zero-shot manner (Zhang et al., 2013), meaning that no transcribed data from the target language is required. A simple approach is to perform pattern matching at the acoustic level, usually requiring a variant of dynamic time warping (DTW). However, the advent of unsupervised methods for neural network based acoustic word embedding raises the potential that such embeddings could be used for QbE, or even more sophisticated IR tasks for languages without a written form, or even for languages whose speakers would benefit from voice interfaces but where speech transcription tools are unreliable.

In this paper, we leverage speech embeddings similar to Sanabria et al. (2023a) and introduce novel inference techniques designed specifically for IR with unwritten languages. We conduct a comprehensive set of development experiments in which we compared the technique to common competing methods – including both DTW and a discrete string search – on a QbE proxy task that we create for several Indic languages. We go on to evaluate the method on a recently-collected corpus of Gormati (Reitmaier et al., 2024), an actual unwritten language. This data was collected specifically with IR in mind, and enables us to evaluate the performance of our method against metrics that are directly relevant for the community in question. The main contributions of this work are that to our knowledge, this is the first successful approach on a real-world IR task for an unwritten language; and our method is the first to develop speech embeddings for a new language in a zero-resource setting targetted specifically to SDR.

2 Prior work

2.1 Spoken Document Retrieval

The key challenges of SDR are how to represent spoken queries and documents, and how to perform search using those representations. Most approaches use large vocabulary continuous speech recognition (LVCSR) to transcribe both queries and documents into text, followed by standard text IR methods (Chelba et al., 2008). In cases where high recall is required, lattices containing alternative candidate transcriptions can be used in place of a single 1-best transcription (James and Young, 1994; Richardson et al., 1995).

Historically, SDR was performed without the need for word-based transcription by using phonetic transcriptions instead (Amir et al., 2001; Ng and Zue, 2000). This approach was commonly termed "phonetic search". When both queries and documents are transcribed into sequences of discrete phoneme-like symbols, it is possible to use string-matching algorithms to perform keyword search. However, the matching must be robust to the high error rates typically seen in phone recognition, requiring methods such as Buzo et al.'s (2013) windowed string search method – which calculates string distances between queries and segments of documents – or retrieval with the vector space model (VSM), using phone n-grams as terms (Moreau et al., 2004). It should be noted that phonetic search methods often exhibit very high false positive rates.

When performing QbE or another form of fully speech-speech retrieval, it is also possible to perform matching with continuous representations in the acoustic domain. In this case, it is essential to perform dynamic time warping to account for the differing term lengths between query and document audio. Early work used standard signal processing features such as mel-frequency cepstral coefficients (MFCCs) (Park and Glass, 2008), but such features are not robust to variation in speaker charactertistics or acoustic environment (Sudhakar et al., 2023). Subsequently many alternative neural-network features have been tested, including phone posteriorgrams (Hazen et al., 2009) and multilingual bottleneck features (BNFs) (van der Westhuizen et al., 2022). San et al. (2021) found that self-supervised features from wav2vec 2.0 and XLSR-53 can outperform MFCCs and BNFs using DTW.

For low-resource languages, LVCSR systems may suffer from unacceptably high error rates, or may not be available at all; and of course, for unwritten languages it simply may not be possible to produce word-like output. In such cases, it may be necessary to use phonetic search methods or acoustic domain matching. We compare both of these approaches in our experiments.

2.2 Acoustic Word Embeddings

Acoustic Word Embeddings (AWEs) are embeddings of speech that aim to capture word-like properties. In theory, they may be able to use contextual information to learn semantic information, in a manner similar to text-based word embeddings. Compared to text, however, speech data has a much higher time resolution; contains additional nuisance factors that are unrelated to word identity; and is generally available in more limited quantities. Furthermore, word boundaries are generally unknown. However, since they can be trained in an unsupervised manner on a target language – or trained on related languages – AWEs can be useful for untranscribed languages (Sanabria et al., 2023a; Jacobs and Kamper, 2021).

The extent to which AWEs are able to capture semantic information is still a current research topic. Pasad et al. (2024) demonstrate that self-supervised representations (e.g., HuBERT vectors) do contain some level of semantic information, useful for discriminating words. They additionally show that when these features are used as inputs to downstream models, they perform much better at word discrimination than with other more standard features - e.g., MFCCs.

Pasad et al. (2024) show that pooling self-supervised representations (e.g., from wav2vec2 or HuBERT) can produce effective AWEs, and Sanabria et al. (2023b) find HuBERT to be the best for English word discrimination. Although, since HuBERT is only trained on English, the quality degrades when it is applied to other languages. However, the recent release of mHuBERT, a compact model with the same architecture as HuBERT-Base, trained on 147 languages, could enable generating high-quality AWEs for languages beyond English

(Boito et al., 2024; Hsu et al., 2021).

Instead of pooling, one can train a model that uses self-supervised representations to produce AWEs. Sanabria et al. (2023a) describe a simple method of producing AWEs using a learned pooling layer trained on at least one hour of target language speech. They use a multilingual phone recogniser (MPR) to transcribe the recordings and then train the model contrastively to embed speech segments with the same transcription similarly. The limitations of this method are that it is not clear how to apply it to a QbE task and it requires at least one hour of target language training data plus an MPR. In contrast, Hu et al. (2021) and Jacobs and Kamper (2021) explored the performance of transfer learning with AWE models on a QbE task by training on well-resourced languages and applying the models to low-resource target languages without finetuning. Both studies embedded the entire query, segmented the search collection with a sliding window and embedded these segments. Jacobs and Kamper (2021) found that training using languages that are closely related to the target language improves performance, and when adding training languages, the largest improvement is gained from adding a single related language. We hypothesise that adopting this approach with a learned pooling model, with its lower data requirements, could allow for the development of an effective QbE system using an AWE model trained with only a small amount of related language training data.

3 AWE Model for QbE

Our approach, which we refer to as the AWE model, uses learned pooling over mHuBERT representations to acquire AWEs. We also introduce two inference methods that use these AWEs to perform retrieval.

3.1 mHuBERT Model

One simple way to perform the QbE task is to use mHuBERT vectors as-is. We first convert document and query recordings into sequences of vectors by directly passing them through mHuBERT. Then, we compute cosine similarity between all extracted vectors for a given query and document. Finally, for each query vector, the maximum similarity over the document vectors is taken and the similarities are averaged over the query vectors to get a single similarity value between a query and a document, illustrated by Figure 1. This is done for all combinations of queries and documents and for each query, the documents are ranked based on their similarity scores.

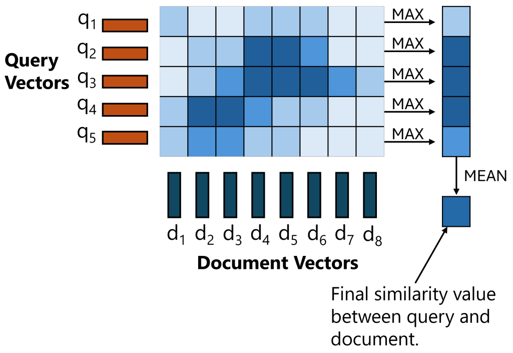
q_1
q_2
q_3
q_4
q_5
Query Vectors
MAX
MAX
MAX
MAX
MAX
MEAN
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8
Document Vectors
Final similarity value between query and document.

Figure 1: Example of calculating the final similarity value between a query and a document from the cosine similarity matrix. Darker cells indicate higher similarity.

However, mHuBERT vectors only cover very short, fixed-length segments (20 ms) making it difficult to capture information from longer, variable-length words (Pasad et al., 2024; Algayres et al., 2022). For more word-like representations, we can combine multiple mHuBERT vectors together through pooling methods.

3.2 Learned Pooling

To build on the vanilla mHuBERT model, we obtain AWEs by learning a pooling function over mHuBERT features. The pooling function is trained using the NTXent contrastive loss as in Sanabria et al. (2023a). As input, this loss takes a batch consisting of several pairs, each from a different class. Within each pair, the two examples serve as positive examples for each other, while examples from other pairs act as negatives, and vice versa for the other pairs. Samples are selected based on their phonetic transcriptions – segments that share the same transcription are considered positive samples. We train using gold phone transcriptions as the training language can be higher resource than the target language and so may have gold transcriptions. For when gold labels are unavailable, we experiment with MPR transcriptions.

3.3 AWE Model Inference

We present two inference methods for SDR with AWEs based on a sliding window approach, similar to those discussed in Section 2.2. However, here it is necessary to window both the document and the query because for Gormati, the queries can be just as long or longer than many of the documents.

The first method, *Phone Window Inference* (Figures 2 and 4), relies on the phone timings from an MPR to divide recordings into segments of continuous, non-silence phones.[1] The min/max length of these segments is specified in phones. For example, for 2-4 phones, all segments containing 2 continuous, non-silent phones are extracted first, followed by extracting segments with 3, and 4 phones. These segments are then embedded using the AWE model, and queries and documents are compared using the cosine distance method, described in Section 3.1.

The second method, *Time Window Inference* (Figures 3 and 4), does not require phone timing knowledge. Instead, an average phone length is assumed and the window is applied as a standard sliding window with 50% overlap. E.g., with an 80 ms average phone length, 2-4 phones would equal windows of 160 ms, 240 ms, and 320 ms.

We anticipate that phone window inference with gold labels will outperform time window inference and phone window inference with an MPR, because of the additional noise from silences and partial phones in the time window and from incorrect transcriptions with the MPR. However, we treat this is a top line system, since gold labels would not be available at inference time in a deployment scenario.

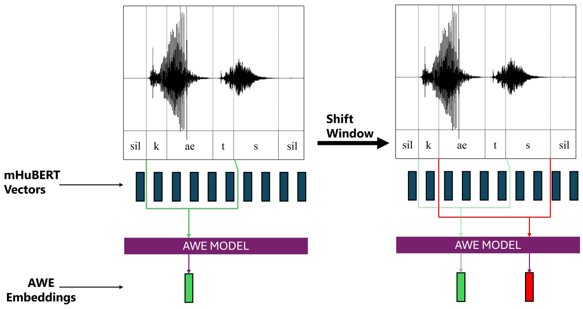

Figure 2: Example phone window inference with a length of 3.

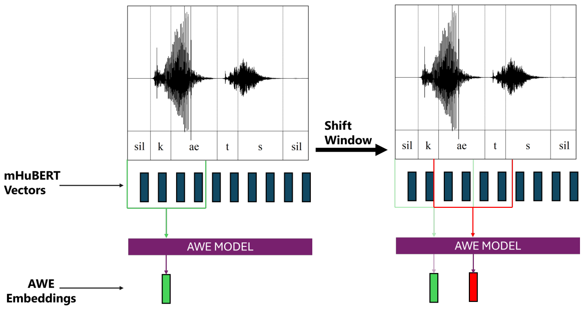

Figure 3: Example of time window inference.

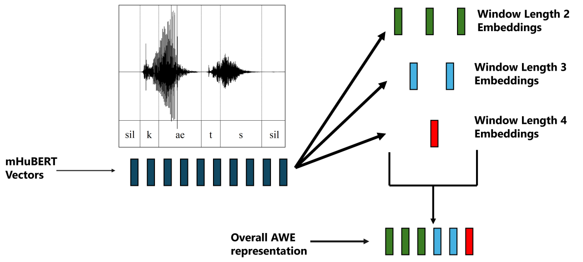
Window Length 2 Embeddings
Window Length 3 Embeddings
Window Length 4 Embeddings
sil
k
ae
t
s
sil
mHuBERT Vectors
Overall AWE representation

Figure 4: Example AWE representation for a recording using either phone or time window inference with a length of 2-4.

4 Data

4.1 Gormati Dataset

Our primary task is IR for Gormati, an unwritten language spoken by the Banjara farming community in India. This dataset was recently collected by Reitmaier et al. (2024). Community members were asked to provide a natural spoken description of images of various crops. The dataset contains 302 recordings (3.8 hours) split over 32 different classes/images.

To select Gormati queries, we removed silent recordings and those longer than 3 minutes to avoid memory issues. Any classes with only 1 recording were removed from the corpus. The remaining recordings were divided into queries and documents. We used 99 queries as in Reitmaier et al. (2024). The queries were unmodified recordings randomly selected from a given class, and the number of queries in each class was proportional to the number of recordings in that class. We ensured that classes with only 2 recordings had at least 1 query. This left 288 documents and 99 queries consisting of natural spoken descriptions. Since these descriptions might discuss a topic indirectly rather than directly naming the subject, there is no assurance of any lexical or phonetic overlap between a query and its corresponding documents.

Our processed data had small discrepancies with the data described in Reitmaier et al. (2024), which we were unable to reconcile despite best efforts. However, their search collection was restricted to only include high volume classes, while ours has no such restriction, including classes with as few as two recordings; hence, our formulation should be more difficult and realistic.

[1] Note that, for training and inference, recordings are first passed through mHuBERT before they are split up into different segments.

Language	# Queries	Corpus Size
Gormati	99	288
Gujarati	896	23,255
Hindi	163	4,686
Marathi	89	2,550
Odia	30	873
Tamil	983	28,321
Telugu	984	28,504

Table 1: Data corpus sizes and number of queries.

4.2 Indic Datasets

Given the limited Gormati data, we used higher-resource Indic language data during the development of our models. We used data for Gujarati, Hindi, Marathi, Odia, Tamil, and Telugu from the 2021 Interspeech Multilingual and Code-Switching (MUCS) challenge (Diwan et al., 2021). For each language, we combined the training and test sets to form the search corpus, we filtered out short utterances under 4 words, and we sampled one example of each repeated sentence. Data not used for the search corpus was used for training.

For QbE, we extracted single-word queries using tf-idf weighting. We imposed a minimum document frequency of 2 and a maximum of 6. We ranked each word by its maximum tf-idf value and selected the top scorers as queries, such that the ratio of queries to corpus size was 0.03-0.04, as in Table 1. For each keyword, we selected one recording as the query source, while the remaining recordings containing the keyword were the corresponding gold standard matches.

5 Methods

5.1 Baseline Implementation

To gauge the performance of the AWE model, we chose traditional DTW acoustic matching and phone recognition-based search methods as our baselines, as mentioned in Section 2.1. We implemented DTW using mHuBERT representations (3rd iteration, final layer) (Boito et al., 2024) as features. For each query, we ranked relevant recordings based on normalised subsequence DTW (Giorgino, 2009; Tormene et al., 2009) with Euclidean local distance.[2]

For the phone-based matching baselines, we tested both VSM retrieval and windowed string search. We used the MPR from Reitmaier et al. (2024) to transcribe both queries and documents, then performed search on these transcriptions.

For VSM retrieval, we represented queries and documents as vectors of tf-idf weighted terms and scored based on their cosine similarity. We used all phone n-grams from 1-grams to 8-grams as terms.

For approximate string search, we slid a window of 1.2 times the query length over each document. Documents were scored based on their edit distance within the window, and documents with the lowest scores were returned as matches.

5.2 mHuBERT Model

The most important consideration for mHuBERT is what layer to extract the representations from. We used layer 9, which we found through experimentation to be optimal. See Appendix A for results over more layers.

5.3 AWE Model

The architecture of the pooling function is the same as in Sanabria et al. (2023a) and Algayres et al. (2022), with a layer norm followed by a 1D convolution, then by a transformer layer with positional embeddings, and finally by a max pooling layer through time (total: 6.8M params).

We train three monolingual models separately on 2 hours of Tamil, Telugu and Gujarati using the NTXent contrastive loss.[3] During training, we test the multilingual search performance of models at each epoch by testing on a Marathi search task. We early stop when performance does not improve for 2 epochs. We use the Adam optimiser with learning rate, $l = 10^{-4}$ and we set the NTXent temperature $\tau = 0.07$. Training takes under 5 hours on an NVIDIA V100 16GB (Volta).

For time window inference, we assume an average phone length of 80 ms. For phone window inference, we use the MPR from Reitmaier et al. (2024).

5.4 Hyperparameters

Following initial experiments, we determined optimal hyperparameters for model training of: 9, 0.07, and 10^{-4} for layer, temperature and learning rate, respectively.

For inference: 3-9 phones was optimal for both time and phone (gold and MPR) window inference for the MUCS languages. For Gormati: 4-13

[2] To embed a MUCS query, we first embed the whole recording containing the query, then extract the series of vectors representing the query using gold standard timings.

[3] We choose these languages since they are sampled at 16 kHz, the required sample rate for mHuBERT.

phones and 3-7 phones was optimal for time and phone (MPR) window inference, respectively.

For MUCS languages, phone window (MPR) marginally outperformed time window, so we use MPR phone window inference (3-9) with MUCS languages. For Gormati, time window inference outperformed phone window (MPR), so we use time window inference (4-13) with Gormati. Discussion of these results is continued in Section 6.2.

5.5 Evaluation

Reitmaier et al. (2024) evaluated their Gormati voice search system with a *Top 5* metric, which is the percentage of queries that had at least one correct document in their top 5 returns. They used this metric because their voice search app displayed 5 images per page and users could reliably identify a single correct image among them. We use this metric partly for consistency, but also because our systems could be integrated into a similar user-facing application in future. However, the Top 5 metric is quite coarse - it does not consider the number of results in the top 5 or their order. Likewise, it does not indicate how the system performs across all returns. Hence, in addition to the Top 5 metric, we use Mean Average Precision (MAP) and Mean Average Precision at 5 (MAP@5).

6 Results

6.1 Baseline Results

Results for the traditional baseline methods are reported in Table 2. We found that DTW consistently outperformed both types of retrieval using the MPR. These results suggest that the MPR transcriptions were simply too inconsistent for even our approximate string retrieval methods.

We tested the accuracy of the MPR on one of our MUCS test languages, Tamil. We transcribed all Tamil queries with the MPR and treated these as “reference" transcriptions. Then, we transcribed all instances of query words within documents and quantified the mismatch between these and the “references" using phone mismatch rate (PMR).[4] The results highlighted the MPR’s poor performance, revealing a PMR of 54%. We also examined how well the MPR can detect voice activity. Using the 2 hours of Tamil training data and comparing it to the gold labels, we determined there were 89 minutes of voice activity. Nevertheless, we use these results as baselines for comparison to the AWE model.

Additionally, we report results for the mHuBERT model in Table 3 over all languages.

6.2 AWE Model Initial Experiments

As shown in Section 5.4, the optimal inference lengths for Gormati differ to that for the MUCS languages. The Gormati time window length (4-13) is much longer than that for the MUCS languages (3-9). This could be because Gormati queries are generally much longer (average 34 s) than our MUCS language queries (average <1 s), meaning longer phone sequences occur more frequently and therefore may be more discriminative. In contrast, the Gormati phone window (MPR) length (3-7) is similar to that for the MUCS languages (3-9). This could be since the MPR is inaccurate, regularly deletes phones and inserts silences, meaning long phone sequences are less likely to occur and those that do occur are unlikely to be transcribed correctly.

6.3 Ensemble Models

We hypothesized that by ensembling models trained on different languages we may produce a model that performs well over all metrics and over all languages. To ensemble models, we simply averaged the scores for each document, for each query, over all models (training languages: Gujarati, Tamil and Telugu, as in Section 5.3). Results in Table 4 show that the ensemble model performs well over all metrics. See Appendix B for results for non-ensemble models.

Considering inference methods again, Table 5 shows that for the ensemble model, phone window (MPR) inference performs on average slightly better than time window inference for the MUCS languages, matching our initial results. A full breakdown is in Appendix C. Table 5 additionally shows that the performance of phone window (MPR) inference is much lower than the top-line results with the gold labels. This highlights the importance of a good MPR and demonstrates that performance can still be enhanced by improving the MPR. Furthermore, in contrast to the MUCS languages, Table 5 shows that on Gormati, time window inference performs much better than phone window (MPR) inference. This could indicate that the MPR performs much worse on Gormati than other languages.

6.4 Training with MPR Labels

Training with MPR-predicted phone timings removes the requirement for labelled training data,

[4] Phone mismatch rate is similar to phone error rate.

Language	DTW			MPR VSM			MPR String Search		
	Top 5	MAP@5	MAP	Top 5	MAP@5	MAP	Top 5	MAP@5	MAP
Gujarati	44.0%	0.332	0.312	22.8%	0.163	0.148	19.9%	0.143	0.138
Hindi	43.6%	0.328	0.333	24.5%	0.167	0.177	27.0%	0.190	0.196
Marathi	61.8%	0.484	0.401	34.8%	0.265	0.201	28.1%	0.214	0.184
Odia	73.3%	0.517	0.432	30.0%	0.232	0.196	53.3%	0.367	0.304
Tamil	46.6%	0.357	0.340	12.7%	0.090	0.085	13.6%	0.089	0.086
Telugu	40.7%	0.307	0.286	18.9%	0.133	0.118	20.0%	0.142	0.131
Average	*51.7%*	*0.388*	*0.351*	*24.0%*	*0.175*	*0.154*	*27.0%*	*0.191*	*0.173*

Table 2: Baseline DTW and MPR results for MUCS languages.

Language	Top 5	MAP@5	MAP
Gormati	68.7%	0.496	0.228
Gujarati	48.1%	0.214	0.215
Hindi	50.3%	0.226	0.242
Marathi	64.0%	0.308	0.274
Odia	86.7%	0.367	0.324
Tamil	51.2%	0.401	0.377
Telugu	45.8%	0.199	0.196
Average (MUCS)	57.7%	0.286	0.271

Table 3: Results for the mHuBERT model (layer 9).

Language	Top 5	MAP@5	MAP
Gormati	87.9%	0.683	0.336
Gujarati	62.9%	0.286	0.291
Hindi	60.1%	0.268	0.284
Marathi	69.7%	0.357	0.333
Odia	90.0%	0.410	0.371
Tamil	61.5%	0.479	0.465
Telugu	59.0%	0.270	0.265
Average (MUCS)	67.2%	0.345	0.335

Table 4: Results using the AWE ensemble model, trained with gold labels. MPR phone window inference is used with the MUCS languages and time window is used with Gormati.

Language	Inference	Top 5	MAP@5	MAP
Gormati	MPR	71.7%	0.541	0.254
	Time	87.9%	0.683	0.336
MUCS Average	Gold	75.1%	0.393	0.388
	MPR	67.2%	0.345	0.335
	Time	64.7%	0.338	0.329

Table 5: Results for Gormati and MUCS for the AWE ensemble model, trained with gold labels, with time and phone (Gold and MPR) window inference.

Language	Labels	Top 5	MAP@5	MAP
Gormati	Gold	87.9%	0.683	0.336
	MPR	82.8%	0.660	0.315
MUCS Average	Gold	67.2%	0.345	0.335
	MPR	62.2%	0.325	0.314

Table 6: Results with the AWE ensemble model, trained using MPR/Gold labels. MPR phone window inference is used with the MUCS languages and time window is used with Gormati.

which is useful as low-resource languages often lack labelled data. We expect that training using MPR-predicted phone timings will produce a worse model than using gold timings, due to the added noise from the MPR. However, as the MPR is reasonably effective for inference, we expect that a model trained with MPR-timings could still be reasonably effective.

To test this, we retrained our models using the MPR-predicted timings and compared it to our previous models trained on gold labels. Table 6 shows that the ensembled MPR-trained models are clearly worse than the gold label-trained models, as expected. However, they still perform reasonably well, with metrics that are only at most 7% lower than those of the gold labelled models. These results show that labels are not necessary for building a strong model, and a fully unsupervised transfer learning approach using an MPR can be effective.

6.5 Training with Gormati Data

We hypothesised that training with Gormati (using MPR-predicted labels) could improve performance as we train with the same language we test on. However, based on the results in Section 6.4, it seems that the MPR may not perform well on Gormati. To test this, we finetuned our gold label trained Tamil model on Gormati using files previously excluded from the search collection, totalling around 30 minutes of audio. We used Tamil since it had the highest Top 5 score on Gormati.

We found that finetuning with Gormati degrades model performance. We could have potentially tested further by partitioning additional Gormati data from the search collection and finetuning the model's hyperparameters. However, we chose not to do this since the initial results were very poor and indicated that this method would be unsuccessful.

6.6 Amount of Data

In low-resource contexts, it is useful to know how much data is necessary to train a model effectively. Lower data requirements could enable the use of data from languages that are more closely related to the target language, even if they have less data than other higher-resource but less related languages.

All previous models were trained using 2 hours of data. Here we tested the effect of training using half and a quarter of that amount. We tested using the Tamil model since it had the best Top 5 score on Gormati. We found that reducing the training data to 1 hour produces a very similar model to 2 hours. Further reducing the data to 0.5 hours noticeably impacts performance, though not too drastically. Therefore, in general, increasing the training data increases performance, with the greatest increase between 0.5 to 1 hour of data.

7 Discussion

The best results for each model on the MUCS languages are shown in Table 7. The AWE ensemble model has the best average Top 5 score with 67.2%, though it has a slightly worse MAP and MAP@5 score compared to the DTW baseline. This suggests that the AWE model is much better at producing at least one correct response per query than the DTW model but it is slightly worse when it comes to the overall ranking. Combining these two models could produce a model with high scores over all metrics.

On Gormati, the AWE model performs the best with a Top 5 score of 87.9%, exceeding the best score of 74% from Reitmaier et al. (2024). Note that the DTW model cannot readily be applied to the more complex Gormati document retrieval task. Unlike Reitmaier et al. (2024), our model requires no target language training data and thus can operate on classes that have just one document. Similarly, the success of our transfer learning approach shows that this method can be extended to other low-resource languages using only one hour of data from a related higher-resource language. For best performance, this data must be labelled. However, we showed that a good model can be trained using MPR labels. This shows that a successful model can be produced without any supervised data from the target language, a critical requirement for an unwritten language.

Model	**Top 5**	**MAP@5**	**MAP**
AWE Ensemble	**67.2%**	0.345	0.335
mHuBERT	57.7%	0.286	0.271
DTW Baseline	51.7%	**0.388**	**0.351**

Table 7: Average MUCS language results with a selection of the best-performing models.

The success of the transfer learning approach suggests that the AWEs are mainly capturing phonetic information, as semantic information is unlikely to generalise well across languages. A method that captures more semantic information might produce better results given the nature of the Gormati data. Jacobs and Kamper (2024) present such a method, but it requires knowledge of word boundaries, making it unsuitable for unwritten languages. In the future, adapting this or similar methods to unwritten languages could increase the semantic content of the AWEs, potentially improving information retrieval further.

In a case where there is no data from related languages or resources are unavailable for training, then the mHuBERT or DTW models could be used. They perform worse than the AWE model, but require no training and so can be applied directly to the target language.

8 Conclusion

We have presented a successful unsupervised method for developing a purely speech-based IR system. However, there remain several avenues for future work. Improving our inference method by experimenting with window lengths, strides and overlaps could be valuable. Optimising model architectures could also lead to improvements, as might combining the AWE model with the DTW model. Our model development with the MUCS data was geared towards the specific task of returning documents that directly contained a single-word query. This type of retrieval is insufficient when it is necessary to return semantically similar results to the query, or for multi-word queries that might benefit from partial matching.

9 Limitations

There are several limitations to this study, all of which can be addressed with further work. First, we only experimented with training using Tamil, Telugu and Gujarati, as these were the only related languages where we had approximately similar speech data. However, with additional data, it would be possible to train using other languages more closely related to Gormati, such as Marathi and Hindi. We did not experiment with multilingual training, which could enhance the models' ability to generalise to other languages; neither did we train mHuBERT on related languages to improve the quality of its representations. Our document ranking system was not tuned to the Gormati search task; in future, we could experiment with different similarity metrics and different methods to compare queries and documents. We only experimented with mHuBERT representations but we could experiment with a wider range of self-supervised representations to better determine the optimal representation. We used a somewhat limited number of documents and queries for testing on Odia, Hindi and Marathi; increasing the number of queries and documents would increase our confidence of our results with these languages. Finally, the Gormati dataset used in this work was designed with IR in mind, and was collected collaboratively with members of the Banjara community. When applying methods from this work to other languages, especially low-resource languages, it is important to keep in mind the community being served. This could take the form of catering the system towards a specific application or domain that is most useful to speakers of the target language, or involving speakers in the evaluation process.

Acknowledgments

References

Robin Algayres, Adel Nabli, Benoît Sagot, and Emmanuel Dupoux. 2022. Speech Sequence Embeddings using Nearest Neighbors Contrastive Learning. In *Proc. Interspeech 2022*, pages 2123–2127.

Arnon Amir, Alon Efrat, and Savitha Srinivasan. 2001. Advances in phonetic word spotting. In *Proceedings of the tenth international conference on information and knowledge management*, pages 580–582.

Marcely Zanon Boito, Vivek Iyer, Nikolaos Lagos, Laurent Besacier, and Ioan Calapodescu. 2024. mHuBERT-147: A Compact Multilingual HuBERT Model. *arXiv preprint*. ArXiv:2406.06371 [cs, eess].

Andi Buzo, Horia Cucu, Mihai Safta, and Corneliu Burileanu. 2013. Multilingual query by example spoken term detection for under-resourced languages. In *2013 7th Conference on Speech Technology and Human - Computer Dialogue (SpeD)*, pages 1–6, Cluj-Napoca, Romania. IEEE.

Ciprian Chelba, Timothy J. Hazen, and Murat Saraclar. 2008. Retrieval and browsing of spoken content. *IEEE Signal Processing Magazine*, 25(3):39–49. Conference Name: IEEE Signal Processing Magazine.

Anni R Coden, Eric W Brown, and Savitha Srinivasan. 2002. *Information retrieval techniques for speech applications*, volume 2273. Springer Science & Business Media.

Ana Deumert. 2014. *Sociolinguistics and Mobile Communication*. Edinburgh University Press.

Anuj Diwan, Rakesh Vaideeswaran, Sanket Shah, Ankita Singh, Srinivasa Raghavan, Shreya Khare, Vinit Unni, Saurabh Vyas, Akash Rajpuria, Chiranjeevi Yarra, Ashish Mittal, Prasanta Kumar Ghosh, Preethi Jyothi, Kalika Bali, Vivek Seshadri, Sunayana Sitaram, Samarth Bharadwaj, Jai Nanavati, Raoul Nanavati, and Karthik Sankaranarayanan. 2021. MUCS 2021: Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages. In *Interspeech 2021*, pages 2446–2450. ISCA.

Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software*, 31(7):1–24.

Trina Gorman, Emma Rose, Judith Yaaqoubi, Andrew Bayor, and Beth Kolko. 2011. Adapting usability testing for oral, rural users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1437–1440, Vancouver BC Canada. ACM.

Kira Griffitt and Stephanie Strassel. 2016. The query of everything: developing open-domain, natural-language queries for bolt information retrieval. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3741–3747.

William Hartmann, Damianos Karakos, Roger Hsiao, Le Zhang, Tanel Alumäe, Stavros Tsakalidis, and Richard Schwartz. 2017. Analysis of keyword spotting performance across iarpa babel languages. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5765–5769. IEEE.

Timothy J. Hazen, Wade Shen, and Christopher White. 2009. Query-by-example spoken term detection using phonetic posteriorgram templates. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 421–426.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Yushi Hu, Shane Settle, and Karen Livescu. 2021. Acoustic Span Embeddings for Multilingual Query-by-Example Search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 935–942, Shenzhen, China. IEEE.

Christiaan Jacobs and Herman Kamper. 2021. Multilingual Transfer of Acoustic Word Embeddings Improves When Training on Languages Related to the Target Zero-Resource Language. In *Interspeech 2021*, pages 1549–1553. ISCA.

Christiaan Jacobs and Herman Kamper. 2024. Leveraging multilingual transfer for unsupervised semantic acoustic word embeddings. *IEEE Signal Processing Letters*, 31:311–315.

David A James and Steve J Young. 1994. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages I–377. IEEE.

Nicolas Moreau, Hyoung-Gook Kim, and Thomas Sikora. 2004. Combination of phone n-grams for a mpeg-7-based spoken document retrieval system. In *2004 12th European Signal Processing Conference*, pages 549–552.

Kenney Ng and Victor W. Zue. 2000. Subword-based approaches for spoken document retrieval. *Speech Communication*, 32(3):157–186.

Joseph Olive, Caitlin Christianson, and John McCary. 2011. *Handbook of natural language processing and machine translation: DARPA global autonomous language exploitation*. Springer Science & Business Media.

Alex Park and James Glass. 2008. Unsupervised Pattern Discovery in Speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16:186–197.

Ankita Pasad, Chung-Ming Chien, Shane Settle, and Karen Livescu. 2024. What Do Self-Supervised Speech Models Know About Words? *Transactions of the Association for Computational Linguistics*, 12:372–391.

Thomas Reitmaier, Dani Kalarikalayil Raju, Ondrej Klejch, Electra Wallington, Nina Markl, Jennifer Pearson, Matt Jones, Peter Bell, and Simon Robinson. 2024. Cultivating Spoken Language Technologies for Unwritten Languages. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17, Honolulu HI USA. ACM.

F. Richardson, M. Ostendorf, and J.R. Rohlicek. 1995. Lattice-based search strategies for large vocabulary speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 576–579 vol.1.

Nay San, Martijn Bartelds, Mitchell Browne, Lily Clifford, Fiona Gibson, John Mansfield, David Nash, Jane Simpson, Myfany Turpin, Maria Vollmer, Sasha Wilmoth, and Dan Jurafsky. 2021. Leveraging Pre-Trained Representations to Improve Access to Untranscribed Speech from Endangered Languages. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1094–1101.

Ramon Sanabria, Ondřej Klejch, Hao Tang, and Sharon Goldwater. 2023a. Acoustic Word Embeddings for Untranscribed Target Languages with Continued Pretraining and Learned Pooling. In *INTERSPEECH 2023*, pages 406–410. ISCA.

Ramon Sanabria, Hao Tang, and Sharon Goldwater. 2023b. Analyzing acoustic word embeddings from pre-trained self-supervised speech models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

P Sudhakar, K Sreenivasa Rao, and Pabitra Mitra. 2023. Query-by-Example Spoken Term Detection for Zero-Resource Languages Using Heuristic Search. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Paolo Tormene, Toni Giorgino, Silvana Quaglini, and Mario Stefanelli. 2009. Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11–34.

Ewald van der Westhuizen, Herman Kamper, Raghav Menon, John Quinn, and Thomas Niesler. 2022. Feature learning for efficient ASR-free keyword spotting in low-resource languages. *Computer Speech and Language*, 71(C).

Michael Weintraub. 1993. Keyword-spotting using sri's decipher large-vocabulary speech-recognition system. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 463–466. IEEE.

Yaodong Zhang et al. 2013. *Unsupervised speech processing with applications to query-by-example spoken term detection*. Ph.D. thesis, Massachusetts Institute of Technology.

A mHuBERT Model Layer Results

Table 8 shows results for the mHuBERT model over layers 7-11. It shows that layer 9 is the best over all metrics with layer 8 trailing closely. The other layers appear noticeably worse than these two.

Layer	Top 5	MAP@5	MAP
7	53.3%	0.282	0.237
8	57.2%	0.313	0.261
9	**59.3%**	**0.316**	**0.265**
10	50.4%	0.264	0.216
11	39.8%	0.207	0.163

Table 8: Average metrics for the mHuBERT model for all languages (MUCS and Gormati), for various layers.

B Single System AWE Results

We trained our AWE models on three languages: Tamil, Telugu, and Gujarati. The results for these non-ensemble models are shown in Table 9 for each test language. MPR phone inference is used for MUCS languages and time window inference is used for Gormati.

C AWE Ensemble Model Results

Table 10 contains a breakdown of the results for the ensemble model, trained on gold labels over different inference methods and MUCS test languages.

Language	**Tamil training**			**Telugu training**			**Gujarati training**		
	Top 5	**MAP@5**	**MAP**	**Top 5**	**MAP@5**	**MAP**	**Top 5**	**MAP@5**	**MAP**
Gormati	88.9%	0.670	0.310	85.9%	0.696	0.336	85.9%	0.686	0.343
Gujarati	57.7%	0.261	0.265	58.4%	0.261	0.268	63.5%	0.286	0.289
Hindi	55.2%	0.244	0.260	56.4%	0.251	0.267	62.0%	0.274	0.286
Marathi	65.2%	0.334	0.314	65.2%	0.333	0.314	66.3%	0.340	0.307
Odia	80.0%	0.383	0.350	80.0%	0.376	0.339	73.3%	0.368	0.350
Tamil	59.7%	0.456	0.442	59.4%	0.451	0.436	59.8%	0.454	0.439
Telugu	54.3%	0.251	0.246	57.3%	0.263	0.259	55.8%	0.257	0.252
Average (MUCS)	*62.0%*	*0.332*	*0.313*	*62.8%*	*0.323*	*0.314*	*63.5%*	*0.330*	*0.321*

Table 9: Results on each test language (using MPR phone inference for MUCS languages and time window inference for Gormati) for AWE models with different training languages. The average is only shown over MUCS languages.

Language	**Phone Window (Gold)**			**Phone Window (MPR)**			**Time Window**		
	Top 5	**MAP@5**	**MAP**	**Top 5**	**MAP@5**	**MAP**	**Top 5**	**MAP@5**	**MAP**
Gujarati	69.8%	0.315	0.319	62.9%	0.286	0.291	60.6%	0.274	0.277
Hindi	70.6%	0.315	0.332	60.1%	0.268	0.284	59.5%	0.277	0.289
Marathi	79.8%	0.397	0.377	69.7%	0.357	0.333	70.8%	0.352	0.337
Odia	93.3%	0.473	0.450	90.0%	0.410	0.371	76.7%	0.388	0.362
Tamil	71.1%	0.562	0.553	61.5%	0.479	0.465	61.2%	0.469	0.452
Telugu	66.1%	0.293	0.296	59.0%	0.270	0.265	59.2%	0.266	0.259
Average	*75.1%*	*0.393*	*0.388*	*67.2%*	*0.345*	*0.335*	*64.7%*	*0.338*	*0.329*

Table 10: Results for ensemble model for different inference methods.