# Leveraging spoken term detection for improved resource access for language documentation and revitalisation projects

**Nay San**
Department of Linguistics
Stanford University
nay.san@stanford.edu

## Abstract

Nearly half of languages spoken today are considered endangered and there are many ongoing efforts to record remaining speakers of these languages. As recording audio is much easier than transcribing recorded audio, language documentation efforts tend to yield large amounts of untranscribed audio, which is difficult to index and search. In this work, we investigate how access to untranscribed audio can be improved using query by example spoken term detection (QbE-STD). We extend recent work on QbE-STD using convolutional neural networks (CNNs) and additionally test these CNNs on language documentation data from two Australian Aboriginal languages, Kaytetye and Warumungu. Results showed that the CNNs outperformed the baseline system based on Dynamic Time Warping (DTW) and at a promising level of performance for use in language documentation projects.

## 1 Introduction

Of the estimated 7,000 languages in the world today nearly half of them may no longer exist after a few more generations. There are thus many ongoing efforts to document remaining speakers of endangered languages and to help revitalise such languages by developing language learning materials. For a number of these efforts, how easily recorded language resources may be used by all interested parties — from language teachers, to community members, to linguists — is directly impacted by how straightforwardly these resources can be searched and retrieved.

This project aims to provide an experimentally-informed evaluation and discussion of how access could be improved for one particular type of language resource — untranscribed speech. Untranscribed speech is particularly problematic for search and retrieval in language documentation contexts as information retrieval systems are typically text-based and there is rarely the volumes of already-transcribed material necessary to train a speech-to-text system with a sufficient level of accuracy for practical use.

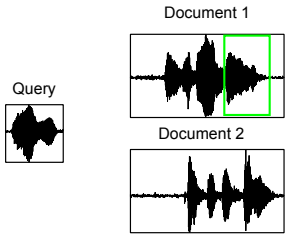


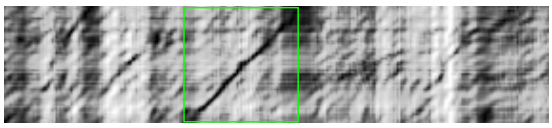

Figure 1: An illustration of Query-by-Example Spoken Term Detection (QbE-STD)

In this project, we investigate to what extent access to untranscribed audio in language documentation corpora can be improved through the use of query-by-example spoken term detection (QbE-STD). QbE-STD is defined as the task of finding all regions within a set of audio documents in which a spoken query term occurs. Figure 1 below illustrates the task where a query term (e.g. 'coffee') is searched in a corpus of two reference documents. The term is correctly detected at the end of Document 1 (e.g. 'I had some coffee'), as shown by the green box, while none are detected in Document 2 (e.g. 'It is rainy today').

## 2 Related work

State-of-the-art approaches to QbE-STD typically use an iterative Dynamic Time Warping (DTW) approach, where a window the size of the query is moved along the reference document and a DTW-based distance score is calculated at each step. Depending on the size of the corpus and number of queries to be searched, this method can be computationally expensive. Nevertheless, DTW has consistently been shown to be hard to beat and provides a competetive baseline, e.g. [1], [2].

a) True positive (query occurs in document)



b) True negative (query does not occur in document)

Figure 2: Distance matrices between spectral features of a reference document and two different queries. Occurence of a query is associated with a quasi-diagonal band indicating a high degree of spectral and temporal correlation.

An alternative approach based on convolutional neural networks (CNNs) has recently been proposed in [1], [2]. In this approach, the task of QbE-STD is reformulated as an image classification problem. As shown above in Figure 2, the occurrence of a spoken query within a reference document is associated with the presence of a quasi-diagonal band (representing a high degree of spectral and temporal correlation) in the distance matrix between the features of the spoken query and those of the reference document.

Results from [1] showed that the CNN-based approach outperformed the DTW baseline across all languages in the Spoken Web Search (SWS2013) benchmark dataset. This dataset includes a wide variety of languages, including several 'low-resource' African languages (isiXhosa, isiZulu, Sepedi, Setswana). However in [2], a later study by the same authors on the same datasets in which multilingual bottleneck features were used instead of the phone posteriors in [1], the DTW baseline was mostly on par with the CNN-based method and outperformed both the CNN and end-to-end methods for the lower resource languages in the SWS2013 dataset [2, Fig. 9].

In this project, we perform a similar set of CNN- vs. DTW-based QbE-STD system comparisons as in [1], [2]. In place of training bottleneck feature extractors from scratch, we use an off-the-shelf bottleneck feature extractor [3].[1] In addition, we also examine the performance differences between the CNN architecture proposed in [1], [2] (henceforth 'Ram2018') and other standard image classification architectures such as VGG and ResNet. We test these systems on language documentation corpora from two Australia Aboriginal languages and also test the Ram2018 model trained for this project on the SWS2013 dataset to provide a point of comparison with prior work.

## 3 Datasets

Table 1 below provides an overview of the datasets used in this project. Note for SWS2013 setup, both the development and evaluation queries are searched on the same corpus (i.e. of 10,726 files shown below). Following [1], [2], 495 of the 505 queries (1,510 tokens) are used for training the CNNs. For the training phase, true negatives were randomly sampled at each epoch to provide a 1:1 ratio of positive and negative examples. The remaining 10 of 505 dev queries were held out as a Training-Dev set. For the current project, we use the Warumungu Picture Dictionary (WPD) as the development set. The two tests sets (I and II) of interest are the Kaytetye Picture Dictionary (KPD; single speaker) and the Kaytetye Learner's Guide (KLG; multiple speakers). Both are language learning resources for Kaytetye with accompanying audio, which are representative data for the use cases of interest for this project. Finally, to compare our Ram2018 model to those in previous work, we also test on the SWS2013 evaluation data (Test III).

Table 1: Descriptive statistics of data used in project

| | Train (CNNs) | Training-dev (CNNs) | Dev (CNNs, DTW) | Test (CNNs, DTW) | | |
|---|---|---|---|---|---|---|
| | | | | I | II | III (Ram2018-only) |
| Dataset | SWS2013-dev | SWS2013-dev | WPD | KPD | KLG | SWS2013-eval |
| Query set | 1,510 | 10 | 383 | 397 | 157 | 503 |
| Corpus size | 10,762 | 10,762 | 383 | 397 | 809 | 10,762 |
| True positives & negatives ratio | 26,366:26,366 (1:1) | 147:27,989 (1:190) | 1,130:145,559 (1:129) | 1,042:157,609 (1:151) | 619:127,013 (1:205) | 5,562:5,408,016 (1:972) |

[1] Available on https://speech.fit.vutbr.cz/software/but-phonexia-bottleneck-feature-extractor

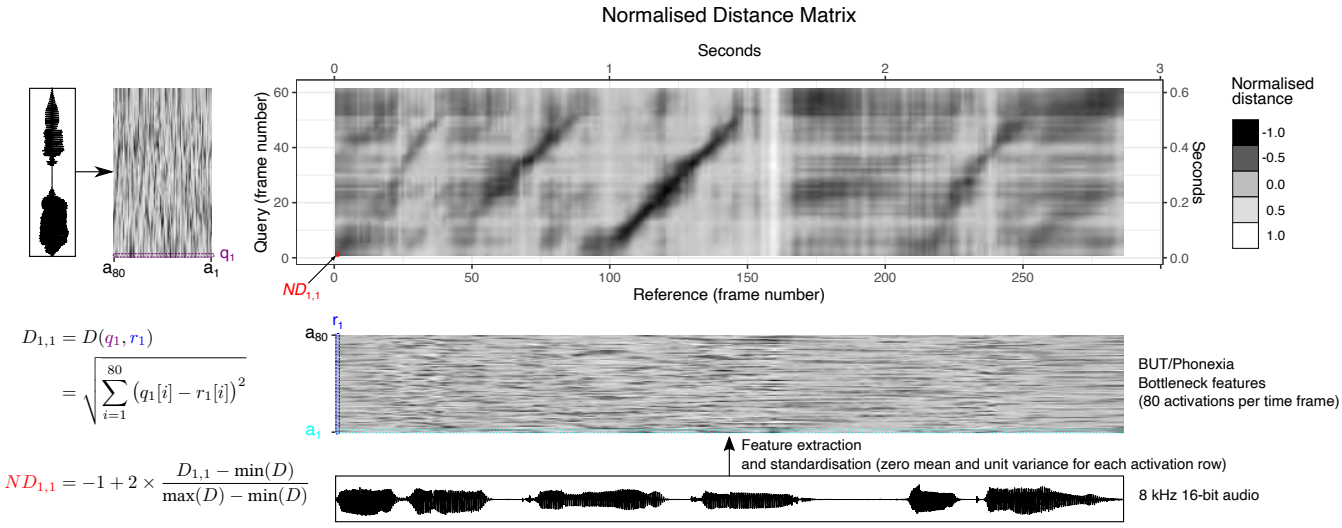



Figure 3: Process for constructing the normalised distance matrix from query and reference audio files.

## 4 Evaluation metric

We use $F_2$ ($F_\beta$ with $\beta = 2$) as the evaluation metric for experiments reported here. Formally defined below in (1), $F_\beta$ provides an interpretable way of weighting recall over precision (when $\beta > 1$) or vice versa (when $\beta < 1$); i.e. $F_\beta = F_1$ when both are equally weighted. Two common choices for $\beta$ are 0.5 (precision twice as important) and 2 (recall twice as important).

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}} \quad (1)$$

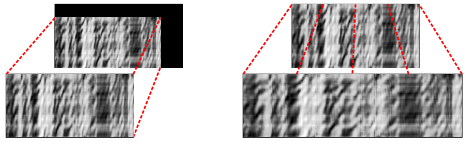
While the most appropriate $\beta$ value(s) should be verified with user experience testing in future studies, it is clear that a relatively low precision system with moderate-to-good recall is more useful for our use case (i.e. with $\beta > 1$). Users are unlikely to be searching on untranscribed audio had searches on transcribed audio already yielded sufficient results of interest. Thus, the system should try to retrieve as many potentially relevant documents as possible, even if many are false positives. At the same time, a relatively high false positive rate in the system is also likely tolerable as this helps discover and index similar sounding terms which are nonetheless useful in language documentation settings.

## 5 Distance matrix construction

We now describe the process for constructing a distance matrix between a query term and reference audio document illustrated above in Figure 3. First, given the audio files for a query $Q$ and a reference $R$, features corresponding to 80 activation values per time frame are extracted using the BUT/Phonexia bottleneck features extractor. This extraction process yields two feature matrices $F_Q$ of shape $(80, M)$ and $F_R$ of shape $(80, N)$, where $M$ and $N$ are the lengths of the query and reference, respectively.

The normalised distance matrix is calculated from these feature matrices using the standardised Euclidean distance. First, values within each activation component (i.e. rows) in each feature matrix are standardised to have zero mean and unit variance in order to eliminate scale differences between components. The distance matrix $D$ of shape $(M, N)$ is then constructed by calculating the Euclidean distance between each feature vectors $q_1, ..., q_M$ in $F_Q$ and $r_1, ..., r_N$ in $F_R$ (i.e. columns in each feature matrix). To allow for comparisons across different combinations of queries and references, values in $D$ are range normalised to [-1, 1], yielding the normalised distance matrix $ND$ illustrated above in Figure 3.

Given that fixed-size inputs are required for a CNN, matrices of shape $(100, 800)$ are derived from normalised distance matrices of shape $(M, N)$ by padding or evenly sampling along each dimension as appropriate. For example, as shown below in Figure 4a, when both $M < 100$ and $N < 800$, the values of the distance matrix are padded with -1 (the minimum value for normalised distances). On the other hand, when $M = 100$ and $N > 800$ for example as in Figure 4b, the columns of the distance matrix are evenly sampled.



a) Padding b) Even sampling

Figure 4: Deriving fixed-size inputs from variably-sized normalised distance matrices.

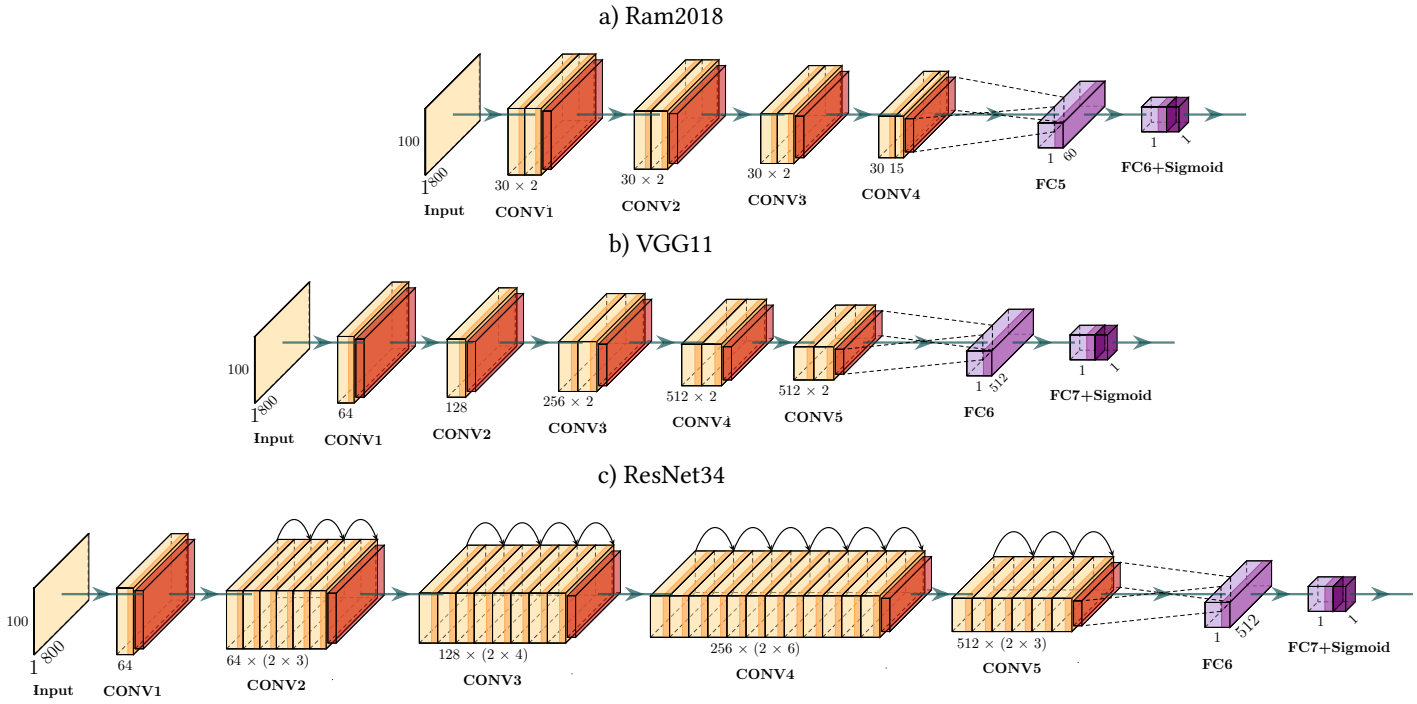



Figure 5: Illustration of the neural network architectures explored in this project.

## 6 Neural network architectures and training

The neural network architectures explored in this project are illustrated above in Figure 5. The Ram2018 network in Figure 5a is that proposed in 2018 by Ram et al. who note that the design was informed by "the VGG network which has been shown to perform well in [the] image recognition task" [1, p. 93]. Accordingly, we also explore the QbE-STD performance of a more complex VGG model such as VGG11 as shown in Figure 5b. Finally, we also explore the performance of a residual network (ResNet34, as shown in Figure 5c), which has superceded VGG networks in image recognition tasks.

To avoid implementational errors, the code for these networks were adopted from existing codebases with appropriate modifications. For Ram2018, this was the repository associated with [1],[2] For VGG11 and ResNet34, this was the TorchVision models in the PyTorch repository.[3] For VGG and ResNet, the modifications needed were to the number of channels in the input (from 3 to 1), to the input sizes for the first fully connected (FC) layer (to work with CNN ouputs given 100 x 800 inputs), and the number of output classes (from 1000 for ImageNet to 1 for binary classification). The PyTorch code for all three networks is available on the GitHub repository associated with this project.[4]

Aside from the learning rate, all three networks were trained using the same procedure. For Ram2018, the learning rate of 0.001 was used. For VGG11 and ResNet34, a learning rate of 0.0001 was used as upon initial experimentation training losses did not decrease with 0.001. The Adam optimization algorithm was used to optimise binary cross entropy loss. The mini-batch size was set to 20 and batches of negative examples were sampled at each epoch. The networks were trained for 50 epochs with checkpoints every 5 epochs, when performance on the Training-Dev and Dev data were also evaluated.

## 7 Results and discussion

The best performing models on the WPD (Dev) dataset according to their $F_2$ scores are displayed below in Table 2.

Table 2: Model peformance on the dev dataset (WPD)

| Model (Epoch) | $F_2$ | Precision | Recall |
|---|---|---|---|
| VGG11 (10) | **0.580** | 0.388 | **0.662** |
| ResNet34 (10) | 0.575 | **0.526** | 0.589 |
| Ram2018 (5) | 0.547 | 0.366 | 0.624 |
| DTW (n.a.) | 0.378 | 0.217 | 0.465 |

Note that the best model, VGG11, outperforms both the DTW baseline and the Ram2018 model in both precision and recall. While the ResNet34 model attains the highest precision, it appears to do so at the cost of recall, thus resulting in an overall lower $F_2$.

[2]https://github.com/idiap/CNN_QbE_STD
[3]https://github.com/pytorch/vision/tree/master/torchvision/models
[4]See: https://github.com/fauxneticien/bnf_cnn_qbe-std/blob/main/src/Models.py

Table 3: Model performance on the test datasets.

| Model | $F_2$ | |
|---|---|---|
| | I: KPD | II: KLG |
| VGG11 | **0.553** | **0.295** |
| ResNet34 | 0.534 | 0.281 |
| Ram2018 | 0.503 | 0.266 |
| DTW (baseline) | 0.365 | 0.093 |

Figure 6: Precision-recall curves of model performance on the test datasets.

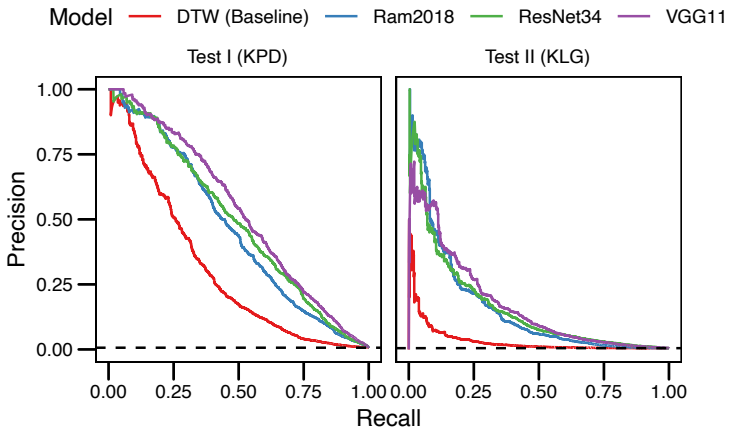



Table 3 displays the results of evaluating the best performing models discussed above on the test datasets, KPD and KLG. For both these test sets, VGG11 again attains the highest $F_2$ score. Further, VGG11 consistently outperforms the other models across various thresholds, as indicated in Figure 6 (purple line consistently closest to top-right corner).

Between the tests sets, there is a consistent drop in performance across all models, where higher performance is seen on KPD than KLG. The fact that KPD is a single-speaker dataset and that KLG is a multi-speaker dataset appears to account for some of this difference. For example, when predictions of the VGG11 model on KLG were analysed in two separate subsets (same speaker in query and reference vs. different speakers in query and reference), the highest $F_2$ score for the same speaker subset was 0.351 while for the different speakers subset it was 0.268.

Finally, we also tested the Ram2018 model trained for this project on the SWS2013-eval dataset in order to compare its performance with results previously reported in [1], [2]. To facilitate this comparison, we generated Maximum Term Weighted Values (MTWVs) using the same parameters (cost of false alarm: 1, cost of missed detection: 100) as in the prior work. Our Ram2018 model achieves a MTWV of 0.517, higher than the range of 0.3986 – 0.4115 reported in [2, Table V]. This increase in performance appears to be compatible with increases associated with additional languages in training the bottleneck feature extractor reported in [2, Table III]. While [2] trained their own bottleneck feature extractor from scratch on a maximum of 5 European languages, this project used the pre-trained BUT/Phonexia extractor which was trained on a typologically-diverse set of 17 languages [3].[5]

## 8 Conclusion and future work

In this project, we examined the performance of three CNN architectures and a baseline DTW system on the task of QbE-STD on datasets from two Australian Aboriginal languages. Replicating previous work, all CNN systems were found to outperform the DTW baseline. Extending previous work, the VGG11 architecture was shown to further outperform the network architecture previously proposed (Ram2018).

The best performing models being found relatively early in the training process (Epochs 5 - 10) suggests that all three model architectures are quickly overfitting to the training data. While the SWS2013-eval dataset was held out in this project to allow for comparison with previously reported models, the SWS2013-eval dataset as well as those from other QbE-STD benchmark datasets could be added to straightforwardly increase the size of the training data in future work.

## Acknowledgements

Thanks to Samantha Disbray and Myf Turpin for making the Warumungu and Kaytetye data available, and Dhananjay Ram for insightful correspondance about his prior work. Also thanks to CS230 TAs Jo Chuang and Shahab Mousavi, and faculty advisor Chris Manning for helpful input about project direction.

## References

[1] D. Ram, L. M. Werlen and H. Bourlard, 'CNN Based Query by Example Spoken Term Detection.,' in *INTERSPEECH*, 2018, pp. 92–96.

[2] D. Ram, L. Miculicich and H. Bourlard, 'Neural Network based End-to-End Query by Example Spoken Term Detection,' *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1416–1427, 2020.

[3] A. Silnova, P. Matejka, O. Glembek, O. Plchot, O. Novotnỳ, F. Grezl, P. Schwarz, L. Burget and J. Cernockỳ, 'BUT/Phonexia Bottleneck Feature Extractor.,' in *Odyssey*, 2018, pp. 283–287.

[5] Cantonese, Pashto, Turkish, Tagalog, Vietnamese, Assamese, Bengali, Haitian Creole, Lao, Tamil, Zulu, Kurdish, Tok Pisin, Cebuano, Kazach, Telugu, Lithuanian.