MULTILINGUAL BOTTLENECK FEATURES FOR QUERY BY EXAMPLE SPOKEN TERM DETECTION

Dhananjay Ram, Lesly Miculicich, Hervé Bourlard

Idiap Research Institute, Martigny, Switzerland
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ABSTRACT

State of the art solutions to query by example spoken term detection (QbE-STD) rely on bottleneck feature representation of the query and audio document. Here, we present a study on QbE-STD performance using several monolingual as well as multilingual bottleneck features extracted from feed forward networks. In contrast to previous works, we use multitask learning to train the multilingual networks which perform significantly better than the concatenated monolingual features. Additionally, we propose to employ residual networks (ResNet) to estimate the bottleneck features and show significant improvements over the corresponding feed forward network based features. The neural networks are trained on GlobalPhone corpus and QbE-STD experiments are performed on a very challenging QUESST 2014 database

***Index Terms*—** Multilingual feature, Bottleneck feature, Residual network, Multitask learning, Query by example spoken term detection

1. INTRODUCTION

Query-by-example spoken term detection (QbE-STD) is the task of detecting audio documents from an archive, which contain a spoken query provided by a user. In contrast to textual queries in keyword spotting, QbE-STD requires spoken queries enabling a language independent search without the need of a full speech recognition system. The search is performed in the acoustic feature domain without any language specific resources, making it a zero-resource task.

The QbE-STD systems primarily involve the following two steps: (i) extract acoustic feature vectors from both the query and the audio document and (ii) employ those features to compute the likelihood of the query occurring somewhere in the audio document as a sub-sequence. Different types of acoustic features have been used for this task: spectral features [1,2], posterior features (posterior probability vector for phone or phone-like units) [3,4,5] as well as bottleneck features [6,7]. The matching likelihood is generally obtained by computing a frame-level similarity matrix between the query and each audio document using the corresponding feature vectors and employing a dynamic time warping (DTW) [3,6] or convolutional neural network (CNN) based matching technique [8]. Several variants of DTW have been used: Segmental DTW [1,9], Slope-constrained DTW [10], Sub-sequence DTW [11], Subspace-regularized DTW [12,13] etc. Subspace detection of posterior features using sparse recovery has also been used for frame level query detection [4,12,14]. State of the art performance has been achieved using bottleneck features with DTW [6].

Bottleneck features [15,16,17] are low-dimensional representation of data generally obtained from a hidden bottleneck layer of a feed forward network (FFN). This bottleneck layer has a smaller number of hidden units compared to the size of other layers. The smaller sized layer constrains information flow through the network which enables it to focus on the information that is necessary to optimize the final objective. Bottleneck features have been commonly estimated from auto-encoders [15] as well as FFNs for classification [16]. Language independent bottleneck features can be obtained using multilingual objective function [17].

In this work, we present a performance analysis of different types of bottleneck features for QbE-STD. For this purpose, we train FFNs for phone classification using five languages to estimate five distinct monolingual bottleneck features. We also train multilingual FFNs using multitask learning principle [18] in order to obtain language independent features. We used a combination of three and five languages to analyze the effect of increasing the language variation for training.

Previous studies have shown the effectiveness of convolutional neural network (CNN) for acoustic modeling in speech recognition [19, 20]. Residual networks (ResNet) is a special kind of CNN which is effective for learning deeper architectures and has been shown to be very successful for image classification [21] as well as speech recognition [22,23]. This inspired us to use ResNets instead of FFNs to estimate monolingual and multilingual bottleneck features for QbE-STD. To the best of our knowledge, this is the first attempt to use ResNets for bottleneck features estimation.

In the rest of the paper, we present the multitask learning approach used to train the multilingual networks in Section 2.

The research is funded by the Swiss NSF project 'PHASER-QUAD', grant agreement number 200020-169398.

Then, we explain the monolingual and multilingual architectures using FFNs and ResNets in Sections 3 and 4 respectively. Later, we describe the experimental setup in Section 5, and we evaluate and analyze the performance of our models using QUESST 2014 database in Section 6. Finally, we present our conclusions in Section 7.

2. MULTITASK LEARNING

Multitask learning [17, 18] have been used to exploit similarities across tasks resulting in an improved learning efficiency when compared to training each task separately. Generally, the network architecture consists of a shared part and several task-dependent parts. In order to obtain multilingual bottleneck features, we model phone classification for each language as different tasks, thus we have a language independent part and a language dependent part. The language independent part is composed of the first layers of the network which are shared by all languages forcing the network to learn common characteristics. The language dependent part is modeled by the output layers (marked in orange in Figures 1 and 2), and enables the network to learn particular characteristics of each language. In the following sections we present different architectures that we use to obtain the multilingual bottleneck features as well as monolingual ones for comparison.

3. FEED FORWARD NETWORKS

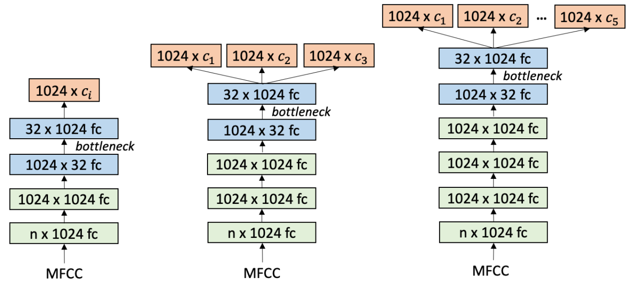
Feed forward networks have been traditionally used to obtain bottleneck features for speech related tasks [6, 16, 17]. Here, we describe the different architectures employed in this study as shown in Figure 1:

(a) Monolingual: our monolingual FFN architecture, consists of 3 fully connected layers of 1024 neurons each, followed by a linear bottleneck layer of 32 neurons, and a fully connected layer of 1024 neurons. The final layer feeds to the output layer of size c_i corresponding to number of classes (e.g. phones) of the i-th language.

(b) Multilingual (3 languages): this architecture consists of 4 fully connected layers having 1024 neurons each, followed by a linear bottleneck layer of 32 neurons. Then, a fully connected layer of 1024 neurons feeds to 3 output layers corresponding to the different training languages. The 3 output layers are language dependent while the rest of the layers are shared among the languages.

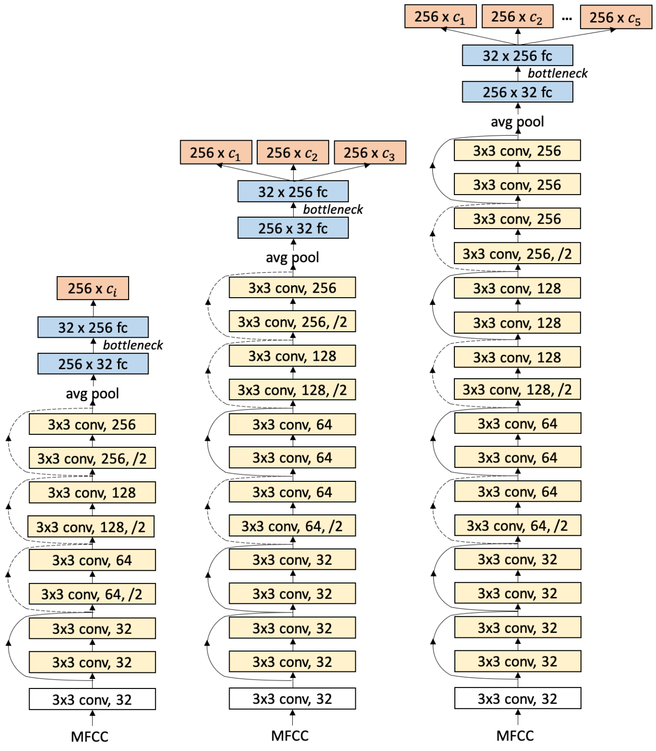
(c) Multilingual (5 languages): this architecture is similar to the previous one except it uses an additional fully connected layer of 1024 neurons, and two extra output layers corresponding to the 2 new languages.

The increased number of layers are intended at modeling the extra training data gained by adding new languages.



(a) Monolingual (b) Multilingual: 3-lang (c) Multilingual: 5-lang

Fig. 1. Monolingual and multilingual feed forward network (FFN) architectures for extracting bottleneck features using multiple languages. c_i is the number of classes for the i-th language and n is the size of input vector.



(a) Monolingual (b) Multilingual: 3-lang (c) Multilingual: 5-lang

Fig. 2. Monolingual and multilingual residual network (ResNet) architectures for extracting bottleneck features using multiple languages. c_i is the number of classes for the i-th language. '/2' indicates down-sampling using a convolution layer of stride 2. The dashed shortcut connection is a linear 1×1 convolution layer.

4. RESIDUAL NETWORKS

A Residual Network [21] is a CNN with shortcut connections between its stacked layers. Skipping layers effectively simplifies the training and gives flexibility to the network. Given an input matrix x and an output matrix y, it models the function $y = f(x) + x$ in each stacked layer, where $f(.)$ represents two convolutional layers with a non-linearity in-between. In case the size of the output of f does not match the size of x, one linear convolutional layer is applied to x (implemented using 1×1 convolutions) to match the number of feature maps before the addition operation. Finally, a non-linearity is applied to the summed output y.

Similar to FFNs, we implemented 3 different architectures depending on the number of languages used for training. Those architectures are shown in Figure 2. We use 3×3 filters for all convolution layers throughout the network. Every time we reduce the feature map size by half (using a convolution layer of stride 2), we double the number of filters. Then we perform a global average pooling to obtain 256 dimensional vector. These vectors are passed through a fully connected linear bottleneck layer which feeds to another layer of size 256. This goes to a single or multiple output classes depending on type of network: monolingual or multilingual. Smaller number of layers are used here in comparison to [21] due to the limited amount of training data.

5. EXPERIMENTAL SETUP

In this section, we describe the databases and the preprocessing steps to perform the experiments. Then, we present the details of training different neural networks.

5.1. Databases

GlobalPhone Corpus: GlobalPhone [24] is a multilingual speech database consisting of high quality recordings of read speech with corresponding transcription and pronunciation dictionaries in 20 different languages. In this work, we use French (FR), German (GE), Portuguese (PT), Spanish (ES) and Russian (RU) to train monolingual as well as multilingual networks and estimate the corresponding bottleneck features for QbE-STD experiments. These languages were chosen to have a complete mismatch between the training and test languages, in contrast to previous works [3,6,8,12] when there was partial overlap (e.g. Czech was present in both training and test languages). We have an average of ∼20 hours of training and ∼2 hours of development data per language.

Query by Example Search on Speech Task (QUESST): QUESST dataset [25] is part of MediaEval 2014 benchmarking initiative and is used here to evaluate the performance of different bottleneck features for QbE-STD. It consists of ∼23 hours of audio recordings (12492 files) in 6 languages as search corpus: Albanian, Basque, Czech, non-native English, Romanian and Slovak. The development and evaluation set includes 560 and 555 queries respectively which were separately recorded than the search corpus. The development queries are used to tune the hyperparameters of different systems. There are three types of occurrences of a query defined as a match in this dataset. Type 1: exactly matching the lexical representation of a query, Type 2: slight lexical variations at the start or end of a query, Type 3: multiword query occurrence with different order or filler content between words. (See [25] for more details)

5.2. Neural Networks Training

We use mel frequency cepstral coefficients (MFCC) with corresponding Δ and $\Delta\Delta$ features as input to the neural networks. We chose these features to keep consistency with earlier works [6, 8, 12]. The outputs are mono-phone states (also known as pdfs in Kaldi [26]) corresponding to each language as presented in Section 5.1. The training labels for these networks are generated using GMM-HMM based speech recognizers [27, 28]. The number of classes corresponding to French, German, Portuguese, Spanish and Russian are 124, 133, 145, 130, 151 respectively. Note that, we also trained these networks using tri-phone based senone classes, however they perform worse than the mono-phone based training. All neural network architectures in this work is implemented using Pytorch [29][1].

Feed Forward Networks: The input to the FFNs is MFCC features with a context of 6 frames (both left and right) resulting in a 507 dimensional vector. We apply layer normalization [30] before the linear transforms and use rectifier linear unit (ReLU) as non-linearity after each linear transform except in the bottleneck layer. We train those networks with batch size of 255 samples and dropout of 0.1. In case of multilingual training, we use equal number of samples from each language under consideration. Adam optimization algorithm [31] is used with an initial learning rate of 10^{-3} to train all networks by optimizing cross entropy loss. The learning rate is halved every time the development set loss increases compared to the previous epoch until a value of 10^{-4}. All the networks were trained for 50 epochs.

Residual Networks: We construct the input for ResNet training using MFCC features with a context of 12 frames (both left and right) resulting in a 39×25 size matrix with single channel in contrast to the 3 channel RGB images generally used in image classification tasks. We also conducted experiments by arranging the input MFCC features in 3 channels: static, Δ and $\Delta\Delta$ values [19], however the

[1] https://github.com/idiap/multilingual_bottleneck_features

performance was worse. Batch normalization [32] is applied after every convolution layer and ReLU is used as non-linearity. The networks are trained with batch size of 255 samples and dropout of 0.05 for 50 epochs. We use the same learning rate schedule as the FFNs with initial and final learning rate of 10^{-3} and 10^{-4} respectively.

The number of layers for both FFN and ResNet architectures for different monolingual and multilingual networks are optimized using the development queries to give best QbE-STD performance. The input context size for these networks are optimized as well by varying it from 4 to 14. We observed the optimal context size corresponding to FFN and ResNet are 6 and 12 respectively. We also performed experiments using batch normalization instead of layer normalization for FNN and the other way for ResNet. We noticed that batch normalization yields better performance with convolution layers while layer normalization is better with fully connected layers. The performance gain of the ResNet models over FFN models (as we will see in Section 6) indicates that ResNets are better equipped to capture information from longer temporal context than the FFNs.

5.3. DTW for Template Matching

The trained neural networks are used to estimate bottleneck features for DTW. As a pre-processing step, we implement a speech activity detector (SAD) by utilizing silence and noise class posterior probabilities obtained from three different phone recognizers (Czech, Hungarian and Russian) [33] trained on SpeechDAT(E) database [34]. Those posterior probabilities are averaged and compared with rest of the phone class probabilities to find and remove the noisy frames. Audio files with less than 10 frames after SAD are not utilized for detection experiments, but those are considered during evaluation.

The DTW system presented in [3] is used here to compute the matching score for a query and audio document pair. It utilizes cosine similarity to obtain the frame-level distance matrix from a query and an audio document. This DTW algorithm is similar to slope-constrained DTW [10] where the optimal warping path is normalized by its partial path length at each step and constraints are imposed so that the warping path can start and end at any point in the audio document. The scores generated by the DTW system are normalized to have zero-mean and unit-variance per query in order to reduce variability across different queries [3].

5.4. Evaluation Metrics

Minimum normalized cross entropy ($C_{nxe}^{\min}$) is used as primary metric and maximum term weighted value ($MTWV$) is used as secondary metric to compare performances of different bottleneck features for QbE-STD [35]. The costs of false alarm (C_{fa}) and missed detection (C_m) for $MTWV$ are considered to be 1 and 100 respectively. One-tailed paired-samples t-test is conducted to evaluate the significance of performance improvement. Additionally, detection error tradeoff (DET) curves are used to compare the detection performance of different systems for a given range of false alarm probabilities.

6. EXPERIMENTAL ANALYSIS

In this section, we report and analyze the QbE-STD performance using various bottleneck features estimated from our FFN and ResNet models. Previously, the best performance on QUESST 2014 database was obtained using monolingual bottleneck features estimated using FFNs [6]. We implemented those models to compare with multilingual features as well as corresponding ResNet based models.

6.1. Monolingual Feature Performance

We train five different monolingual networks for both architectures: FFN and ResNet, corresponding to PT, ES, RU, FR, GE languages from GlobalPhone database. We evaluate the features estimated with these networks using QbE-STD as described in Section 5.3. Similar to [6], we did not employ any specific strategies to deal with different types of queries in QUESST 2014. The results are presented using $C_{nxe}^{\min}$ and $MTWV$ metrics in Table 1. We can see that the ResNet based bottleneck features perform better than most of the FFN based features in terms of $C_{nxe}^{\min}$ metric, except for T3 queries with FR, ES and RU features, where the performances are close. We also observe that PT features perform best for both FFN and ResNet models.

6.2. Multilingual Feature Performance

We present the results of our multitask learning based multilingual systems and compare their performance with a simple monolingual feature concatenation approach.

Multitask Learning: We implement two multilingual networks corresponding to each FFN and ResNet architectures discussed in Sections 3 and 4 using 3 languages (PT, ES, RU) and 5 languages (PT, ES, RU, FR, GE). The 3 language network uses the best performing monolingual training languages. Performance of the features extracted from these networks are shown in Table 1. Clearly, ResNet based bottleneck features provide significant improvement over the corresponding FFN based features. We also observe that PT-ES-RU-FR-GE features significantly outperform PT-ES-RU features for both FFN and ResNet model indicating that additional languages for training provide better language independent features.

Feature Concatenation: Another way of utilizing training resources from multiple languages is to concatenate the

Table 1. Performance of the QbE-STD system in QUESST 2014 database using various monolingual and multilingual bottleneck features for different types of evaluation queries. $C_{nxe}^{\min}$ (lower is better) and $MTWV$ (higher is better) is used as evaluation metric.

	Training Language	System	T1 Queries		T2 Queries		T3 Queries	
			$C_{nxe}^{\min}\downarrow$	$MTWV\uparrow$	$C_{nxe}^{\min}\downarrow$	$MTWV\uparrow$	$C_{nxe}^{\min}\downarrow$	$MTWV\uparrow$
Monolingual Feature	Portuguese (PT)	FFN	0.5582	0.4671	0.6814	0.3048	0.8062	0.1915
		ResNet	0.5405	0.4698	0.6607	0.2747	0.7954	0.1802
	Spanish (ES)	FFN	0.5788	0.4648	0.7074	0.2695	0.8361	0.1612
		ResNet	0.5718	0.4465	0.7043	0.2613	0.8465	0.1462
	Russian (RU)	FFN	0.6119	0.4148	0.7285	0.2434	0.8499	0.1385
		ResNet	0.5728	0.4405	0.7017	0.2481	0.8525	0.1346
	French (FR)	FFN	0.6266	0.4242	0.7462	0.2086	0.8522	0.1249
		ResNet	0.5957	0.4225	0.7017	0.2216	0.8540	0.1267
	German (GE)	FFN	0.6655	0.3481	0.7786	0.1902	0.8533	0.1038
		ResNet	0.6389	0.3803	0.7511	0.2230	0.8497	0.1166
Concat. Feature	PT-ES-RU	FFN	0.5450	0.4957	0.6665	0.2985	0.8053	0.1869
		ResNet	0.5072	0.5164	0.6374	0.3162	0.7965	0.1899
	PT-ES-RU-FR-GE	FFN	0.5457	0.4965	0.6715	0.2903	0.8079	0.1930
		ResNet	0.5040	0.5201	0.6309	0.3212	0.7941	0.1914
Multiling. Feature	PT-ES-RU	FFN	0.4828	0.5459	0.6218	0.3626	0.7849	0.2057
		ResNet	0.4554	0.5666	0.6009	0.3529	0.7650	0.2201
	PT-ES-RU-FR-GE	FFN	0.4606	0.5663	0.6013	0.3605	0.7601	0.2138
		ResNet	**0.4345**	**0.5962**	**0.5703**	**0.3815**	**0.7387**	**0.2487**

Table 2. Number of parameters for different mono and multi lingual models using FFN and ResNet architecture.

Model	FFN	ResNet
Monolingual	~ 1.8M	~ 663K
Multilingual: 3-lang	~ 3.1M	~ 1.4M
Multilingual: 5-lang	~ 4.4M	~ 3.0 M

monolingual bottleneck features to perform DTW. We perform two sets of experiments by concatenating monolingual features from PT-ES-RU and FR-GE-PT-ES-RU languages corresponding to both FFN and ResNet. The results are presented in Table 1. We can see that there is marginal improvement over the best monolingual feature (PT) from FFN model, a similar observation was presented in [6]. On the other hand, ResNet based feature (PT-ES-RU) perform significantly better than the corresponding PT features. However, there is no significant performance difference between the ResNet based 3 and 5 language feature concatenation.

We also observe that the multitask learning based features significantly outperform the monolingual feature concatenation, indicating the importance of multitask learning for utilizing training resources from multiple languages. The bold numbers in Table 1 show the best performance for each type of query and performance metric.

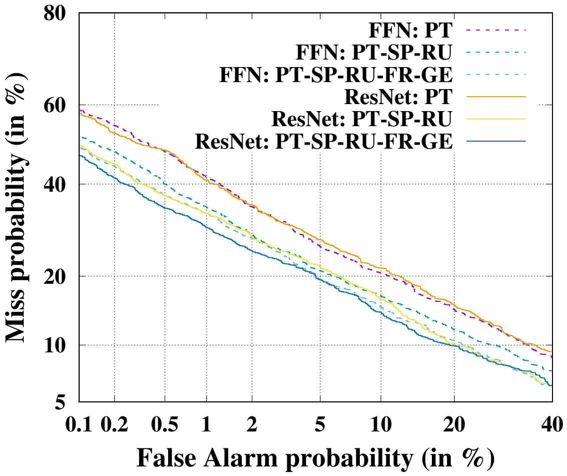


Fig. 3. DET curves showing the performance of monolingual and multilingual features estimated using FFNs and ResNets for T1 queries of QUESST 2014.

The number of parameters for different models are shown in Table 2. We observe that the FFNs have more parameters than the ResNet architectures. The improved performance of ResNet models in comparison to FFNs indicate that ResNet architecture produces better bottleneck features in spite of having less parameters.

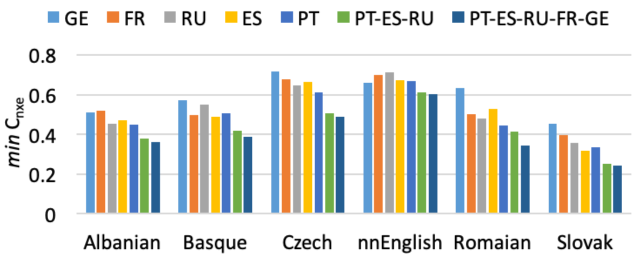


Fig. 4. Comparison of QbE-STD performance of language specific evaluation queries (T1 query) using $C_{nxe}^{\min}$ values

6.3. Monolingual vs Multilingual Feature

The 3 language multilingual feature provides an average absolute gain of 5.2% and 5.8% (in $C_{nxe}^{\min}$) for FFN and ResNet model respectively in comparison to the corresponding best monolingual features (PT). Further 2.3% and 2.5% absolute improvements are observed while using 2 more languages for training. In order to compare the missed detection rates for a given range of false rates we present the DET curves corresponding to these systems in Figure 3. We see a similar trend of performance improvement here as well. We also observe that the performance gain is higher from 1 language to 3 languages than 3 languages to 5 languages. It is due to our use of the best performing languages to train the 3 language network.

6.4. Language Specific Performance

We compare the language specific query performance of ResNet based monolingual and multilingual features as it performs better than the FFN counterparts. We use $C_{nxe}^{\min}$ values of T1 query performance to show this comparison in Figure 4. We observe that the performance improves with more languages used for training, however the amount of improvement varies with language of the query. The smaller performance gain from 3 to 5 languages for some queries (e.g. Albanian, Czech, Slovak) can be attributed to much worse performance of FR and GE features compared to rest of the monolingual features.

7. CONCLUSIONS

We proposed a ResNet based neural network architecture to estimate monolingual as well as multilingual bottleneck features for QbE-STD. We present a performance analysis of these features using both ResNets and FFNs. It shows that additional languages for training improves performance and ResNets perform better than FFNs for both monolingual and multilingual features. Further analysis shows that the improvement is consistent throughout queries of different languages. In future, we plan to train deeper ResNets with more languages to compute and analyze language independence of those features. The improved bottleneck features can be used for other relevant tasks e.g. unsupervised unit discovery. The codes are available at:
`https://github.com/idiap/multilingual_bottleneck_features`

8. REFERENCES

[1] Alex S Park and James R Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[2] Chun-an Chan and Lin-shan Lee, "Model-based unsupervised spoken term detection with spoken queries," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1330–1342, 2013.

[3] Luis Javier Rodriguez-Fuentes, Amparo Varona, Mike Penagarikano, Germán Bordel, and Mireia Diez, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7819–7823.

[4] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard, "Subspace detection of DNN posterior probabilities via sparse representation for query by example spoken term detection," in *Seventeenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016.

[5] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard, "Phonetic subspace features for improved query by example spoken term detection," *Speech Communication*, vol. 103, pp. 27–36, 2018.

[6] Igor Szöke, Miroslav Skácel, Lukáš Burget, and Jan Černockỳ, "Coping with channel mismatch in query-by-example-BUT QUESST 2014," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5838–5842.

[7] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection.," in *INTERSPEECH*, 2016, pp. 923–927.

[8] Dhananjay Ram, Lesly Miculicich, and Hervé Bourlard, "CNN based query by example spoken term detection," in *Proceedings of the Nineteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[9] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 398–403.

[10] Timothy J Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 421–426.

[11] Meinard Müller, *Information retrieval for music and motion*, vol. 2, Springer, 2007.

[12] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard, "Sparse subspace modeling for query by example spoken term detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 6, pp. 1130–1143, June 2018.

[13] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard, "Subspace regularized dynamic time warping for spoken query detection," in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2017.

[14] Dhananjay Ram, Afsaneh Asaei, Pranay Dighe, and Hervé Bourlard, "Sparse modeling of posterior exemplars for keyword detection," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[15] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Twelfth annual conference of the international speech communication association*, 2011.

[17] Karel Veselỳ, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 336–341.

[18] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[19] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[20] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[23] Yu Zhang, William Chan, and Navdeep Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4845–4849.

[24] Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8126–8130.

[25] Xavier Anguera, Luis Javier Rodriguez-Fuentes, Igor Szöke, Andi Buzo, and Florian Metze, "Query by example search on speech at mediaeval 2014.," in *MediaEval*, 2014.

[26] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[27] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[28] Sibo Tong, Philip N Garner, and Hervé Bourlard, "An investigation of deep neural networks for multilingual speech recognition training and adaptation," in *Proceedings of the Eighteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017.

[29] Adam Paszke, Sam Gross, and Soumith Chintala, "Pytorch," 2017, [online] `http://pytorch.org/`.

[30] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[31] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[33] Petr Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Faculty of Information Technology BUT, 2008.

[34] Petr Pollák, Jerome Boudy, Khalid Choukri, Henk Van Den Heuvel, Klara Vicsi, Attila Virag, Rainer Siemund, Wojciech Majewski, Piotr Staroniewicz, Herbert Tropf, et al., "Speechdat (e)-eastern european telephone speech databases," in *the Proc. of XLDB 2000, Workshop on Very Large Telephone Speech Databases*. Citeseer, 2000.

[35] Luis J Rodriguez-Fuentes and Mikel Penagarikano, "Mediaeval 2013 spoken web search task: system performance measures," *n. TR-2013-1, Department of Electricity and Electronics, University of the Basque Country*, 2013.