CNN based Query by Example Spoken Term Detection

Dhananjay Ram, Lesly Miculicich, Hervé Bourlard

Idiap Research Institute, Martigny, Switzerland
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{dhananjay.ram, lesly.miculicich, herve.bourlard}@idiap.ch

Abstract

In this work, we address the problem of query by example spoken term detection (QbE-STD) in zero-resource scenario. State of the art solutions usually rely on dynamic time warping (DTW) based template matching. In contrast, we propose here to tackle the problem as binary classification of images. Similar to the DTW approach, we rely on deep neural network (DNN) based posterior probabilities as feature vectors. The posteriors from a spoken query and a test utterance are used to compute frame-level similarities in a matrix form. This matrix contains somewhere a quasi-diagonal pattern if the query occurs in the test utterance. We propose to use this matrix as an image and train a convolutional neural network (CNN) for identifying the pattern and make a decision about the occurrence of the query. This language independent system is evaluated on SWS 2013 and is shown to give 10% relative improvement over a highly competitive baseline system based on DTW. Experiments on QUESST 2014 database gives similar improvements showing that the approach generalizes to other databases as well.

Index Terms: Deep neural network, Posterior probabilities, Convolutional neural network, Query by example, Spoken term detection, CNN, DTW, QbE, STD

1. Introduction

Query-by-example spoken term detection (QbE-STD) is defined as the task of detecting audio files (from an audio archive) which contain a spoken query. The search is performed relying only on the audio data of query and search space with no language specific resources, making it a zero-resource task. The difference between QbE-STD and keyword spotting is that QbE-STD uses spoken query instead of textual query. Unlike keyword spotting, QbE-STD enables users to search in multilingual unconstrained speech without the help of speech recognition system. It can be viewed as an unsupervised pattern matching problem where the pattern is the information represented by a query.

Different approaches to QbE-STD primarily rely on variations of dynamic time warping (DTW) based template matching techniques [1, 2, 3, 4]. It involves two steps: (i) feature vectors are extracted from the query and test utterance, (ii) these feature vectors are then used to find likelihood score of occurrence. Spectral features [5, 6], posterior features (vectors indicating posterior probabilities for phone or phone-like units) [1, 2] as well as bottleneck features [7] have been used for this task. The posterior features can be extracted from Deep neural network (DNN) [2, 8], gaussian mixture model (GMM) [1], deep boltzman machine (DBM) [9] or using spectral clustering combined with DNN [10].

The feature vectors extracted from both query and test utterance are used to compute a frame level similarity matrix. Several variants of DTW have been proposed to detect a query (which can occur as a sub-sequence) in a test utterance by finding a warping path through the similarity matrix [3]. Segmental DTW [1, 5] constrains the query to match with segments of test utterance. Slope-constrained DTW [11] restricts the slope of the warping path to a certain degree. Sub-sequence DTW [12] enforces the cost of insertion at the beginning and end of the query to be equal to 0. Subspace-regularized DTW utilizes the subspace structure of both query and test utterance to regularize the similarity matrix [4, 13]. Other approaches include hidden markov model (HMM) based symbolic search which relies on unsupervised acoustic units [6, 10, 14]. Sparse recovery based subspace detection method also uses posterior features to perform a frame level query detection [4, 15, 16].

Among the approaches discussed above, DTW with posterior features currently yields state-of-the-art performance. However, the resulting performance levels are still quite limited and not appropriate to real life problems. This limitation, and the recent success of convolutional neural network (CNN) in image classification task [17, 18], motivated us to develop a novel approach to deal with this problem.

Unlike DTW based methods, we view here the similarity matrix as an image and propose to approach the QbE-STD problem as an image classification task. We observe that the similarity matrix contains a quasi-diagonal pattern if the query occurs in the test utterance. Otherwise, no such pattern is observed. Thus for each spoken query, a test utterance can be categorized as an example of positive or negative class depending on whether the query occurs in it or not. This is a straightforward application of CNN for QbE-STD. To the best of our knowledge, it has never been used before. The simplicity of this approach along with significant performance gain makes it very useful for the task.

Tackling the QbE-STD problem as image classification, and exploiting CNN to address this task has the following advantages: (i) CNN provides a learning framework to the problem which is absent in a DTW based system, (ii) CNN considers the whole similarity matrix at once to find a pattern, whereas DTW algorithm takes localized decisions on the similarity matrix to find a warping path, and (iii) CNN based learning also enables a discrimination capability in the system.

In the rest of the paper, we describe the process of image construction in Section 2, and the methodology for classification in Section 3. We evaluate our system on SWS 2013 and QUESST 2014 databases, and analyze the performance in Section 4. Finally, we present the conclusions in Section 5.

2. Image Construction

In this section, we describe the procedure to construct a similarity matrix from a spoken query and a test utterance which is used as an image for binary classification. We follow the same procedure as in [2] to extract posterior feature vectors from both spoken queries and test utterances using pre-trained

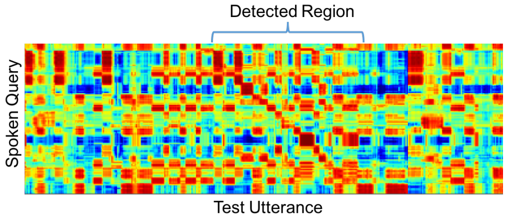

Figure 1: *Positive case: the query occurs in the test utterance*

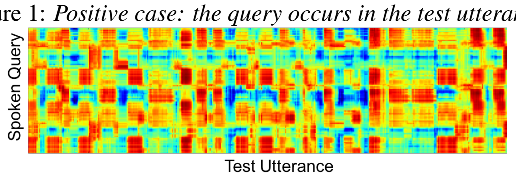

Figure 2: *Negative case: the query does not occur in the test utterance*

feed forward neural networks (see Section 4.3 for more details) with mel frequency cepstral coefficient (MFCC) features as input. Let us consider, $\mathbf{Q} = [\,\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m\,]$ representing the posteriors corresponding to a spoken query and $\mathbf{T} = [\,\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n\,]$ corresponding to a test utterance. Here, m and n represent the number of frames in the query and test utterance respectively. Given any two posterior vectors $\mathbf{q}_i$ and $\mathbf{t}_j$, we compute a distance like measure by taking log of their dot product [1, 5, 11] as follows:

$$s(\mathbf{q}_i, \mathbf{t}_j) = \log(\mathbf{q}_i \cdot \mathbf{t}_j) \quad (1)$$

Higher values of s indicate higher similarity between the vectors. We further apply a range normalization such that all values in the similarity matrix will be between -1 to 1. This helps in dealing with variations in similarity scores for different pairs of query and search utterances.

$$s_{norm}(\mathbf{q}_i, \mathbf{t}_j) = -1 + 2.\frac{(s(\mathbf{q}_i, \mathbf{t}_j) - s_{min})}{(s_{max} - s_{min})} \quad (2)$$

where $s_{min} = min_{i,j}(s(\mathbf{q}_i, \mathbf{t}_j))$, $s_{max} = max_{i,j}(s(\mathbf{q}_i, \mathbf{t}_j))$.
The performance of this similarity score is close to the normalized cosine similarity used in [2], and it is computationally more efficient which is preferable. The similarity matrix is categorized in two class of images: (i) if a query occurs in a test utterance (positive class) and (ii) if a query does not occur in a test utterance (negative class). We present one example for each of this type of images in Figures 1 and 2 respectively. The vertical and horizontal axes represent the frames of query and test utterance respectively. The colors indicate strength of values in the matrix, higher values correspond to red and lower values to blue. The quasi-diagonal pattern observed in the positive class helps to discriminate between the two classes. We present our methodology in the following section to achieve this goal.

3. Methodology

In this section, we present a CNN based classifier for QbE-STD. Our CNN architecture is similar to the VGG network [17] which has been shown to perform well in image recognition task. It consists of a series of convolution and max-pooling layers with a fixed setting of hyper-parameters for all layers, which simplifies the selection of hyper-parameters.

Contrary to the standard image classification task, the input of our CNN is a similarity matrix. Therefore, we use only one channel instead of three corresponding to the RGB color model for images. The architecture consists of four sets of two convolution layers and one max-pooling layer; followed by two fully-connected layers with a soft-max on top. The details are described in Table 1. All convolution layers use ReLU [19] as activation function. The number of channels and dropout were optimized to 30, and 0.2 respectively with a development set. Our architecture has eight convolution layers in total. We expected that a simpler network will be able to perform reasonably well given the simplicity of the task. However, preliminary experiments with less layers were not able to outperform the baseline system. It should be noted that, our system is a language independent system which can be trained using query and test utterance pairs from any language with minimal supervision (without corresponding transcriptions) because it only requires the information whether the query occurs in the test utterance.

Table 1: *CNN Architecture*

Layer	Description
Input	$200\times750\times1$
Conv	Channel: in=1, out=30, Filter: 3x3, Stride: 1
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Maxpool	Channel: in=30, out=30, Filter: 2x2, Stride: 2
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Maxpool	Channel: in=30, out=30, Filter: 2x2, Stride: 2
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Maxpool	Channel: in=30, out=30, Filter: 2x2, Stride: 2
Conv	Channel: in=30, out=30, Filter: 3x3, Stride: 1
Conv	Channel: in=30, out=15, Filter: 3x3, Stride: 1
Maxpool	Channel: in=15, out=15, Filter: 2x2, Stride: 2
FC	Input:$12\times47\times15$, Output=64
FC	Input:64, Output=2
SM	Input:2, Output=2

Conv: Convolution; FC: Fully connected; SM: Softmax

We faced two main challenges to train the CNN for our task which are described as follows:

Variable size input: The similarity matrices have variable widths and lengths corresponding to the number of frames of spoken queries and test utterances respectively. We deal with this issue by fixing the size for all input matrices to an average width and length of the training samples (in our training set, it is 200×750). In case the similarity matrix has length or width larger than the defined input, we down-sample it by deleting its rows and/or columns in regular intervals. On the other hand, if the length or width is smaller, we simply fill the gap with the lowest similarity value from the corresponding distance matrix. Down sampling does not affect the quasi-diagonal pattern severely as the rows and columns being deleted are spread throughout the distance matrix. Also, we did not apply segmentation of test utterances in fixed size intervals because it will require the region of occurrence of the query in a test utterance which is not available for QbE-STD.

Unbalanced data: Typically, the frequency of occurrence of a particular query in the search space is very small. As a consequence, the number of positive and negative samples is highly unbalanced (in our training data is 0.1% to 99.9% respectively). To deal with this problem, we balance the training set with equal number of positive and negative examples. The negative examples were randomly sampled from the corresponding set at each iteration. Preliminary experiments showed that this strategy has better performance than using weighted loss function for training.

4. Experimental Analysis

In this section, we describe the databases and the pre-processing steps to perform the experiments. Then, we present the details of CNN training and analysis of the results.

4.1. Databases

Spoken Web Search (SWS) 2013: We consider the SWS database from MediaEval 2013 benchmarking initiative [20] for training and evaluation of our QbE-STD system. This speech data comes from 9 different low-resourced languages: Albanian, Basque, Czech, non-native English, Isixhosa, Isizulu, Romanian, Sepedi and Setswana. The data was collected in varying acoustic conditions and in different amounts from each language. There are 505 queries in the development set and 503 queries in the evaluation set. Each set consists of 3 types of queries depending on the number of examples available per query: 1, 3 and 10 examples. The corresponding number of queries for development set are 311, 100 and 94, whereas for evaluation set are 310, 100 and 93 respectively. The search corpus consists of $\sim$20 hours of audio with 10762 utterances.

Query by Example Search on Speech Task (QUESST) 2014: We consider QUESST dataset [21] from MediaEval 2014 challenge to evaluate the generalizability of our approach. The search corpus consists of $\sim$23 hours of audio recordings (12492 files) in 6 languages: Albanian, Basque, Czech, non-native English, Romanian and Slovak. The evaluation set includes 555 queries which were separately recorded than the search corpus. We did not use this dataset for training or tuning our model. Unlike SWS 2013 datatset, all queries have only one example available. Besides the 'exact matching' task (Type 1) in SWS 2013, there are two more types of approximate matching tasks in QUESST 2014. Type 2: slight lexical variations at the start or end of a query are considered as match. Type 3: multi-word query occurrence with different order or filler content between words are also considered as match. (See [21] for more details)

4.2. Baseline System

The DTW system with posterior features [2] which gives state of the art performance (without considering the fusion of multiple systems), is used as our baseline system. It uses normalized cosine similarity to compute the distance matrix from a query and a test utterance. The DTW algorithm used here is similar to slope-constrained DTW [11] 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 test utterance.

4.3. Feature Extraction, Pre-processing, Evaluation Metric

We employ the phone recognizers (developed at Brno University of Technology (BUT)) used in the baseline system to extract posterior features. It has three different phone recognizers for Czech, Hungarian and Russian [22] which were trained on SpeechDAT(E) [23] database with 12, 10 and 18 hours of speech respectively. There are 43, 59 and 50 phones for the respective languages. In all cases, 3 additional units were used to model silence and non-speech sounds. The posteriors from all three recognizers are concatenated to obtain the feature vectors for our experiments. These posterior features can be considered as a characterization of instantaneous content of the speech signal independent of the underlying language [2].

We implement a speech activity detector (SAD) following [2] to remove the noisy frames. Any audio file with less than 10 frames after SAD is assigned a default minimum likelihood score without performing any detection experiment. We use these features in both baseline and our proposed system to obtain a likelihood score for each pair of query and test utterance. These scores are normalized to have zero-mean and unit-variance per query, which reduces the variability across different queries and make them comparable for final evaluation [2].

Table 2: *Performance of the DTW [2] and CNN based approach in SWS 2013 for single and multiple examples per query using all evaluation queries. $minCnxe$ (lower is better) and $MTWV$ (higher is better) is used as evaluation metric.*

Examples	minCnxe		MTWV	
	DTW	CNN	DTW	CNN
Single	0.7181	**0.6483***	0.3352	**0.3753***
Multiple	0.6565	**0.6028***	0.3685	**0.3880***

* significant at $p < 0.001$

We use minimum normalized cross entropy ($minCnxe$) as primary metric and maximum term weighted value ($MTWV$) as secondary metric to compare performances of baseline systems with our proposed approach [24]. We consider the cost of false alarm (C_{fa}) to be 1 and cost of missed detection (C_m) to be 100 for computing $MTWV$. We have performed one-tailed paired-samples t-test considering scores per query for significance of results. Additionally, we present detection error tradeoff (DET) curves to compare the detection performance of different systems for a given range of false alarm probabilities.

4.4. CNN Training

The development and evaluation queries in SWS 2013 database share the same search space for QbE-STD. The labels provided for development queries indicate whether a query occurs in a test utterance or not. Thus we only have these queries to train our CNN. We use 495 out of 505 queries for training and rest of the 10 queries are used for tuning which were chosen in a random manner. Effectively, we have 1551 queries when we consider different examples of the same query. We have designed our experiment in this manner to follow the setup of SWS 2013 and make a fair comparison to the best system for this task.

We extract posteriors from all the queries and test utterances, and filter them using a SAD to obtain 1488×10750 training example pairs. Out of these examples 24118 are positive examples and rest are negative examples. We balance the classes following the strategy discussed in Section 3. We combine the examples from both classes and prepare batches of 20 samples of query and search utterance pairs. We use the Adam optimization algorithm [25] with a learning rate of 10^{-3} to train the CNN by optimizing cross entropy loss. The whole setup is implemented using Pytorch [26].

4.5. QbE-STD Performance on SWS 2013

We consider two cases depending on the number of examples per query to evaluate the baseline DTW and our CNN model for QbE-STD. In case of a single example per query, the corresponding posterior features constitute the template. On the other hand, with multiple examples per query we compute an average template using traditional DTW [27] before computing the similarity matrix. For this purpose, we select the example with longest temporal length and find a frame-level alignment of the posteriors using DTW. The posteriors mapped in this manner are averaged together to produce the final template [2]. This process was only performed during test time, however the training samples were formed using only single example per query.

The performance of both systems are presented using $minCnxe$ and $MTWV$ values in Table 2 and corresponding DET curves are shown in Figure 3. In both cases, our system outperforms the baseline system while considering any of the evaluation metrics used. The p-values indicate that the improve-

Table 3: *Comparison of improvement in $minCnxe$ (lower is better) score with multiple examples per query for SWS 2013.*

Examples per query	System	1st Example	All Examples	Relative Improvement
3	DTW	0.7298	0.6682	**8.44%**
	CNN	0.5992	0.5573	7.00%
10	DTW	0.8181	0.6893	**15.74%**
	CNN	0.7581	0.6461	14.77%

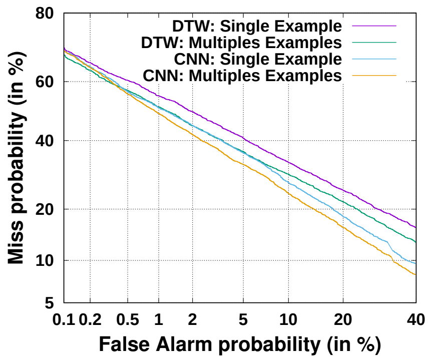

Figure 3: *DET curves showing the performance of baseline DTW and proposed CNN based approach on SWS 2013 for single and multiple examples per query using evaluation queries.*

ments are highly significant. In case of single example, the DET curves show that our system gives lower miss rate for the given range of false alarm rate. While for multiple examples, our system is better than the baseline except for very low false alarm rates.

4.6. Effect of Multiple Examples Per Query

To analyze the effect of introducing multiple examples per query we present a comparison with the baseline system in Table 3. We consider only the queries with multiple examples. We observe that both systems gain with the introduction of more examples per query. The higher gain of the baseline relative to our system can be attributed to the poor performance of the DTW for '1st Example' which gives it more room for improvement. It can also indicate that we need better ways to generate average template from multiple examples than the existing DTW based template averaging method.

4.7. Language Specific Performance

We contrast the language specific performance for our system with the baseline DTW system using $minCnxe$ values in Figure 4. These experiments are performed using a single example per query of the evaluation set. We can see that our system performs better in all cases compared to the baseline system. However, the improvement is marginal in case of 'Isizulu' and 'non-native English'.

4.8. QbE-STD Performance on QUESST 2014

We use the model trained on SWS 2013 for testing on QUESST 2014 evaluation set to analyze the generalizability of our system. We compare our approach to the baseline DTW system [28] for QUESST 2014. Similar to [28], we did not employ any specific strategies to deal with different types of queries in QUESST 2014. The performance of our system along with the baseline system is presented in Table 4. Clearly, our system performs significantly better than the baseline system for all 3 types of queries. The performance gets increasingly worse from Type 1 to Type 2 and from Type 2 to Type 3. This can be attributed to the training of our system which was trained using only Type 1 queries from SWS 2013. However the consistency in performance improvement for all kinds of queries shows that our system is generalizable to newer datasets.

Table 4: *Performance of the baseline DTW [28] and proposed CNN based approach on QUESST 2014 for different types of queries on evaluation set. $minCnxe$ (lower is better) and $MTWV$ (higher is better) is used as evaluation metric.*

Type of Query	minCnxe		MTWV	
	DTW	CNN	DTW	CNN
Type 1	0.4396	**0.3921***	0.5375	**0.5853***
Type 2	0.6407	**0.5162***	0.3276	**0.4159***
Type 3	0.7724	**0.7358***	0.1620	**0.2056***

* significant at $p < 0.001$

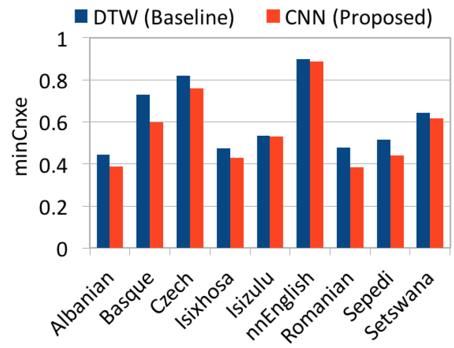

Figure 4: *Comparison of QbE-STD performance of language specific evaluation queries (single example per query) of SWS 2013 using $minCnxe$ values (lower is better)*

5. Conclusions and Future Work

We proposed a novel CNN based approach for QbE-STD. It provides a discriminative learning framework between positive and negative classes, which is not featured in DTW based systems. The performance improvement over baseline system indicates superiority of the new approach. Further analysis shows that the improvement is consistent throughout different languages and databases. However, with multiple examples per query the gain is less than the baseline system indicating the need of further investigation to generate average template. The architecture presented here can be improved with advances in image classification, as well as the use of different input features such as bottleneck. This new approach has the potential to be used in other problems where DTW based systems are applicable (e.g. time series analysis).

Future work includes investigation of better down-sampling and up-sampling techniques to deal with variable size similarity matrices. We also plan to explore end-to-end neural network based system which takes spectral features of query and test utterance as inputs to make a decision, instead of the using a set of pre-trained feed-forward networks for posterior feature extraction. The code is available at:
`https://github.com/idiap/CNN_QbE_STD`

6. Acknowledgements

We would like to acknowledge the authors of [2] for providing their code. The research is funded by the Swiss NSF project 'PHASER-QUAD', grant agreement number 200020-169398.

7. References

[1] Y. Zhang and J. 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.

[2] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. 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.

[3] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, “Spoken content retrievalbeyond cascading speech recognition with text retrieval,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1389–1420, 2015.

[4] D. Ram, A. Asaei, and H. 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.

[5] A. S. Park and J. R. Glass, “Unsupervised pattern discovery in speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[6] C.-a. Chan and L.-s. 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.

[7] H. Chen, C.-C. Leung, L. Xie, B. Ma, and H. Li, “Unsupervised bottleneck features for low-resource query-by-example spoken term detection.” in *INTERSPEECH*, 2016, pp. 923–927.

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[9] Y. Zhang, R. Salakhutdinov, H.-A. Chang, and J. Glass, “Resource configurable spoken query detection using deep boltzmann machines,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5161–5164.

[10] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, “Acoustic segment modeling with spectral clustering methods,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 2, pp. 264–277, 2015.

[11] T. J. Hazen, W. Shen, and C. White, “Query-by-example spoken term detection using phonetic posteriorgram templates,” in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 421–426.

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

[13] D. Ram, A. Asaei, and H. Bourlard, “Subspace regularized dynamic time warping for spoken query detection,” in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2017.

[14] C.-y. Lee and J. Glass, “A nonparametric bayesian approach to acoustic model discovery,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

[15] D. Ram, A. Asaei, P. Dighe, and H. Bourlard, “Sparse modeling of posterior exemplars for keyword detection,” in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[16] D. Ram, A. Asaei, and H. 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.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] X. Anguera, F. Metze, A. Buzo, I. Szoke, and L. J. Rodriguez-Fuentes, “The spoken web search task,” in *the MediaEval 2013 Workshop*, 2013.

[21] X. Anguera, L. J. Rodriguez-Fuentes, I. Szöke, A. Buzo, and F. Metze, “Query by example search on speech at mediaeval 2014.” in *MediaEval*, 2014.

[22] P. Schwarz, “Phoneme recognition based on long temporal context,” Ph.D. dissertation, Faculty of Information Technology BUT, 2008.

[23] P. Pollák, J. Boudy, K. Choukri, H. Van Den Heuvel, K. Vicsi, A. Virag, R. Siemund, W. Majewski, P. Staroniewicz, H. 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.

[24] L. J. Rodriguez-Fuentes and M. 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.

[25] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[26] A. Paszke, S. Gross, and S. Chintala, “Pytorch,” 2017, [online] http://pytorch.org/.

[27] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[28] L. J. Rodríguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, “GTTS-EHU systems for QUESST at mediaeval 2014.” in *MediaEval*, 2014.