Tackling Unseen Acoustic Conditions in Query-by-Example Search Using Time and Frequency Convolution for Multilingual Deep Bottleneck Features

Julien van Hout, Vikramjit Mitra, Horacio Franco, Chris Bartels, Dimitra Vergyri

SRI International, Menlo Park, CA

julien.vanhout@sri.com

Abstract

Standard keyword spotting based on Automatic Speech Recognition (ASR) cannot be used on low- and no-resource languages due to lack of annotated data and/or linguistic resources. In recent years, query-by-example (QbE) has emerged as an alternate way to enroll and find spoken queries in large audio corpora, yet mismatched and unseen acoustic conditions remain a difficult challenge given the lack of enrollment data. This paper revisits two neural network architectures developed for noise and channel-robust ASR, and applies them to building a state-of-art multilingual QbE system. By applying convolution in time or frequency across the spectrum, those convolutional bottlenecks learn more discriminative deep bottleneck features. In conjunction with dynamic time warping (DTW), these features enable robust QbE systems. We use the MediaEval 2014 QUESST data to evaluate robustness against language and channel mismatches, and add several levels of artificial noise to the data to evaluate performance in degraded acoustic environments. We also assess performance on an Air Traffic Control QbE task with more realistic and higher levels of distortion in the push-to-talk domain.

Index Terms: query-by-example, multilingual bottleneck, convolutional neural networks, noise robustness, channel robustness

1. Introduction

Traditional keyword spotting (KWS) systems have relied on building a speech recognition system that detects words, phones or other acoustic units. In data-rich applications, a strong Automatic Speech Recognition (ASR) system generates high-quality word lattices that can be searched for sequences of words of interest [1]. When less linguistic data is available, one can use a phone or syllable recognizer to infer lattices that can be searched for keyword hits. In both cases, out-of-vocabulary words can be dealt with by using automatically inferred pronunciations and approximate search, but some loss in accuracy is expected. Rapid development of a portable and usable KWS system in a new language, dialect, or domain remains a difficult challenge, because training such a system in a way that is usable in potentially channel mismatched and noisy data is heavily reliant on using in-domain training data.

Query-by-Example (QbE) search, where a keyword is enrolled using one or several audio examples, has seen renewed research interest in recent years due to its ability to perform well without necessarily relying on an ASR system. Some QbE systems can even function in a fully language agnostic way, as they do not need knowledge of the languages of interest or language-matched training data, as long as queries are defined either in isolation or in context with precise boundaries. Much progress has been made in recent years thanks to an effort to make languages-agnostic QbE a part of the MediaEval SWS/QUESST evaluations from 2013 to 2015. Work stemming from those evaluations has shown that techniques leveraging supervised, discriminatively trained tokenizers, such as bottleneck features along with dynamic time warping (DTW), are among the highest-performing single systems in language-agnostic QbE in channel- and noise-degraded conditions. Current bottleneck architectures that have been tried for QbE include a simple five-layer bottleneck that can be trained in a multilingual setting [2, 3]. Other, more complex hierarchical architectures learn the bottleneck representation in two steps by first learning a bottleneck whose outputs are then contextualized and fed to a second network that learns improved features for ASR [4, 5]. Such architectures have been used to train monolingual stacked bottleneck systems for QbE in recent MediaEval evaluations [3, 6].

Noise, reverberation, and channel mismatches are the usual causes of speech data mismatches and, hence, are the common sources of performance degradation for ASR systems. While deep neural network (DNN) models have been used in conjunction with noise-robust features to fight channel and noise mismatches, more recently a new type of model called convolutional neural networks (CNN) has been introduced that uses frequency convolution and pooling layers inspired from image recognition. CNNs have been shown to largely outperform standard DNNs in clean speech recognition tasks [7, 8], and these gains were shown to carry over to channel mismatched, noisy, and reverberant speech recognition tasks [8, 9]. In [10], the authors introduced a time-convolution layer parallel to the frequency convolution layer as a way to capture time-scale information and to successfully improve the CNN baseline in reverberant and noisy conditions; this network architecture is referred to as the time-frequency convolutional neural network (TFCNN). CNNs have also started to be used to train noise-robust bottleneck features in other tasks, such as language identification [11].

In this work, we study how CNNs and TFCNNs can be used to train multilingual bottleneck features that are channel- and noise-robust in unseen, mismatched conditions. We show that large improvements in QbE performance are obtained over five-layer DNNs while keeping the network architecture and training very simple. The improvements are shown on the MediaEval QUESST 2014 task where channel mismatch is a challenge, as well as in matched and mismatched noise conditions. We also provide results on a database of real Air Traffic Control (ATC)

This material is based upon work partly supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0037. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. “A” (Approved for Public Release, Distribution Unlimited).

communications to show how these systems transpose in real and challenging acoustic conditions. Besides looking at the influence of bottleneck architecture, we also study the influence of other components such as input feature normalization and speech activity detection (SAD).

2. CNN and TFCNN bottlenecks

This section describes the CNN and TFCNN architectures and how those were used to train and extract bottleneck features.

The CNN has a single convolutional layer with 200 learnt filters of size 8 that perform convolution across the frequency axis, followed by a max-pooling layer with a pooling size of 3. The pooling layer feeds into five hidden layers of size 1024, except for the 3rd layer of size 60 that is used to extract the bottleneck features.

TFCNNs were found to give better performance [10] compared to their CNN counterparts in clean, noisy and reverberated conditions. The TFCNN architecture is shown in Figure 1. It is similar to [10], where two parallel convolutional layers are used at the input, one performing convolution across time, and the other across the frequency axis of the input filterbank features. In our experiments, we used 75 filters to perform time convolution, and 200 filters to perform frequency convolution. For time and frequency convolution, eight bands were used. A max-pooling over three samples was used for frequency convolution, while a max-pooling over five samples was used for time convolution. The feature maps after both the convolution operations were concatenated and then fed to a fully connected neural net, which had 1024 nodes and five hidden layers, where the 3rd hidden layer consisted of a bottleneck with 60 neurons.

We also trained a simple DNN bottleneck baseline consisting of five hidden layers, with 1024 neurons in each layer and a bottleneck in their 3rd hidden layer that consisted of 60 neurons. All networks were trained with cross-entropy and used sigmoid activations for all hidden layers.

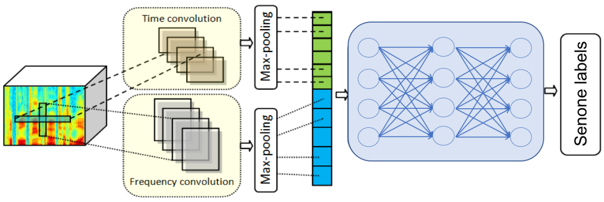

Figure 1: *the TFCNN bottleneck feature extraction pipeline.*

3. QbE system

The proposed query-by-example (QbE) system is similar in architecture to the dynamic-time-warping (DTW) single systems described in [6] and follows a simple architecture consisting of four steps: 1) speech activity detection, 2) bottleneck feature extraction, 3) DTW matching, and 4) score normalization.

3.1. Speech activity detection

As other MediaEval performers showed, speech activity detection (SAD) is crucial to obtaining good QbE performance. Not only is SAD needed to remove potential pauses at the start and end of a query utterance, it can also remove intra-word pauses and other non-speech frames due to channel distortions or background noise. In the type of queries used in MediaEval 2014, where the queries are provided with lose boundaries and are spoken in isolation with a fairly low speaking rate, leading and trailing silences as well as inter-word pauses are especially important to filter out. In other situations, when the query boundaries are precisely set by the user (or obtained from phonetic forced-alignments) and the speaking rate is quite high, SAD may not be as important. In order to assess the impact of SAD, we evaluate two algorithms. Each method generates a SAD score for each frame, and QbE enrollment is performed only using frames above a set threshold, which is optimized for each SAD independently. We tried to apply further smoothing on the SAD scores using median filtering, but the best results were obtained with no smoothing.

3.1.1. DNN-CTS-SAD

This first SAD was developed by SRI under DARPA's RATS (Robust Automatic Transcription of Speech) project. It was trained on 2000 hours of speech from various languages, consisting of clean telephone speech and 7 channels of retransmitted telephone speech from LDC's RATS Speech Activity Detection corpus (LDC2015S02). The scores are produced for each frame by a 3-layer DNN with 500 neurons per layer using contextualized MFCC features with a context of 31 frames. MFCCs themselves are 20-dimensional.

3.1.2. GP-HU-SAD

This SAD is obtained from BUT's Hungarian phone posteriorgram and was trained on the GlobalPhone training set. We use it for SAD in the same way as in [6], by summing the posteriors of all states associated with non-speech phones in order to obtain the final score.

3.2. Bottleneck feature training

The DNN, CNN, and TFCNN bottleneck networks used for feature extraction were trained in a multilingual fashion using speech material originating from five languages and various datasets: Dari (TransTac); Egyptian Arabic (CALLHOME); English (Fisher); Mandarin (GALE); and Spanish (CALLHOME). We used the following Babel data releases: Amharic, IARPA-babel307b- v1.0b; and Pashto, IARPA-babel104b-v0.4bY. FullLP training sets were used. In total, this data comprised approximately 650 hours of audio data in the five languages. All data was sampled at 8 kHz. Note that neither speaker- nor language-level information was ever used in any of the processing outlined in this work. A universal phone set was created by linguistic experts to map phones from all seven languages to a unified set. Acoustic clustering of triphones was then used to create more than 5,000 senones, which were used as targets for the output layer of a bottleneck network. More details regarding the training material are found in [12].

All of our bottleneck systems were trained using time-domain gammatone filterbank (TDGFB) features. The gammatone filters are auditory inspired and have been shown to perform better than traditional filterbanks on tasks where noise and channel mismatches are an issue, either when used directly for ASR [1] or as features for bottleneck training [13]. The TDGFBs were extracted by using SRI's gammatone filterbank implementation, in which a bank of time-domain gammatone filters consisting of 40 channels was used, with the filters equally spaced on the equivalent rectangular bandwidth (ERB) scale. The TDGFBs consisted of filterbank energies computed over an analysis window of 25.6 ms, at a frame advance of 10 ms, with the energies root-compressed using 15th power root followed by an utterance-level mean subtraction across each feature dimension.

To tackle noise and channel mismatch, two different approaches to utterance-level feature normalization were tried on the TDGFB features: spectral mean subtraction (SMS) and spectral mean-variance normalization (SMVN).

3.3. Distance measures

We evaluated several distance measures in order to build the frame-wise distance matrix that we used for DTW search: Euclidean, correlation, cityblock, cosine, dot product, and minus log dot product. Similarly to [6], we found that the cosine distance performed best.

A second dimension that we evaluated was applying transformations to the raw activations from the bottleneck layer, which were extracted before applying the sigmoid. We evaluated several transformations: raw (no transformation), sigmoid, softmax, and 1-sigmoid. We found a consistent gain from using the sigmoid-warped bottleneck activations and therefore use those for the remainder of this paper.

3.4. DTW search

Search was implemented using sub-sequence dynamic time warping (DTW) [14], with the memory-efficient improvements proposed in [15]. The dynamic programming algorithm is succinctly described as follows: we initialized distances to 0 at each frame of the search utterance in order to allow the best paths to start anywhere. Then, we progressively computed the minimum accumulated distance through the joint distance matrix between the query and search utterance. Local path constraints only allowed moving horizontally, vertically, or diagonally by one frame at a time. Path normalization by total path length was applied when making best-path decisions as well as at the end. At each step, the memory-efficient implementation only stored three values: the starting frame, the current path length, and accumulated distance. At the end of the search utterance, we looked for local minima in the normalized accumulated distance of the paths going through the entire query ending at each frame. For each local minimum, we retrieve the stored starting frame for this path. Pairwise comparison of all detections for a particular query enabled merging the detections overlapping by more than 50% by keeping the detection of least normalized distance.

3.5. Score normalization

Since the normalized distance D_{norm} of any path lies in [0,1], we can map the normalized distance to a detection score S as follows:

$$S = 1 - D_{norm}$$

When plotting these distributions for each query, one finds that although unimodal, the means and variances are highly dependent on each query. Also, because those distributions are not quite Gaussian, but rather actually have a longer tail towards lower scores, using z-normalization is not an optimal way to normalize scores across queries. We tried a simple rank normalization, which maps each distribution to a uniform distribution, but found that the $m\text{-}norm$ procedure proposed in [6] provided slightly better performance on all metrics of the evaluation data.

4. Evaluation Data

In this section, we describe the two benchmarks that were used to run QbE experiments. First, we describe the multilingual cross-channel MediaEval QUESST 2014 task, and how we added artificial noise and reverb to this dataset. Second, we introduce a new QbE benchmark based on English Air Traffic Control speech.

4.1. QUESST 2014

We evaluated our system on the QUESST (QUery by Example Search on Speech Task) track of the MediaEval 2014 evaluation [16]. This task is particularly interesting because of the many languages involved and the channel mismatches introduced by using of various corpora and recording conditions. We enrolled using the set of 307 dev queries that qualified as "T1" queries during the evaluation, meaning that they needed to exactly match the reference. Approximate matching queries "T2" and "T3" were not the focus of this study. All queries were telephone recorded and do not overlap with the corpora used in testing, simulating a real-world scenario of channel, speaker and domain mismatch. Each word in a query had at least four phonemes, and no other information (e.g., language tag) was provided. We tested on the 23 hours of audio that comprise the search corpus. More details, as well as a breakdown of the different languages, is be found in Table 1.

Table 1: *The QUESST 2014 evaluation setup consists of six European languages. We evaluated only on DEV queries with exact matching (T1 queries)*

Language	Search (minutes)	Search Domain	Num. Queries
Albanian	127	read	20
Basque	192	broadcast	16
Czech	237	conversational	77
NN-English	273	TEDx	46
Romanian	244	read	46
Slovak	312	parliamentary	102
Total (dev)	1385	mixed	307

The primary metric for this evaluation was C_{nxe}, which measure the information brought in by the system as well as the quality of calibration of the scores. A C_{nxe} of 0 shows that the system has perfect information, while a C_{nxe} of 1 or more shows a non-informative or mis-calibrated system. Since this metric measures score calibration and we did not focus on calibration for this work, we only report $minC_{nxe}$ which is computed over the best-possible linear transformation of our system's scores for this set.

To assess the performance of our approach under noisier conditions than provided in the above dataset, we added noise to each of the query and search waveforms. Fourteen different noise types (such as babble, factory noise, traffic noise, highway noise, crowd noise, etc.) were added with a signal-to-noise ratio (SNR) between 10 to 80 dB. These new enrollment and test conditions are referred to here as $noisy1$. In addition, we wanted to assess robustness against noise types different than those used in the noisy data set mentioned above. These consisted of more non-stationary animal noises such as cricket chirping, dog barking, etc., and were added with a SNR between 10 to 60 dB to the query data only. This enrollment condition is referred to as $noisy2$. This test set was created to assess the generalization capability of the proposed QbE approach presented in this work. The original MediaEval data is referred to as $clean$. For this study, we evaluated both matched and mismatched enrollment and test scenarios, with clean queries used to enroll and tested against noisy search data, and vice-versa.

4.2. Air Traffic Control

We use the Air Traffic Control (ATC) database [17] to provide a second benchmark for QbE in more realistic and challenging acoustic conditions. This database includes English radiowave voice communication traffic between various controllers and pilots at three US airports, and was initially released as a benchmark for ASR under LDC catalog number LDC94S14A.

We split the data into two parts: the first part was used for query enrollment while the second was used for search. A total of 167 frequent query n-grams were selected (e.g. ”join the localizer”, ”maintain three thousand”, etc.), and forced alignment was used to obtain word-level alignments for enrollment and ground-truth to evaluate QbE search. Each query had an average of 20 examples available for enrollment, but only one example per query was used for enrollment within the scope of this paper. Follow-up work will evaluate the impact of using multiple queries. The nature of extracting queries in-sentence from automatic alignments leads to queries with no leading or trailing silences.

The original utterances from the ATC data were either spoken by tower controllers (32 distinct speakers) or by airline pilots (283 distinct speakers). While both have a significant degree of degradation and acoustic variation, the latter is of particularly low degree of intelligibility due to the distance between the plane and the receiver, engine noise, weather, non-native speech, breathing from the pilot into the microphone, etc. Also, due to the nature of Air Traffic conversations, the rate-of-speech is quite high and intra-word pauses are very short to non-existent throughout this corpus. In the following sections, we refer to three different evaluation scenarios as follows:

ATC test 1 Enroll and test on controller speech

ATC test 2 Enroll and test on pilot speech

ATC test 3 Enroll on pilot speech, test on controller speech

These will allow to evaluate how algorithms respond to varying degrees of distortions and enrollment/test mismatches.

5. Results and Discussion

In this section, we present the results of our proposed QbE system using TFCNN bottlenecks compared to baselines using DNN and CNN bottleneck features. We also present the effects of using different normalizations on the feature side in order to fight mismatch between training, query enrollment, and search. Finally, we present results assessing the effect of SAD on the two QbE benchmarks.

5.1. Bottleneck architecture and normalization

Results for the original and renoised experiments using the MediaEval 2014 database are shown in Figure 2 for the DNN and TFCNN systems, as well as the impact of using SMS and SMVN normalization. In the clean/clean condition, which is similar to the T1 dev track of the QUESST 2014 evaluation, we show that using a TFCNN bottleneck architecture achieved a significant improvement over a simple DNN bottleneck, as $minC_{nxe}$ was reduced by 15.3% absolute (from 58% to 42.7%) in the SMVN case.

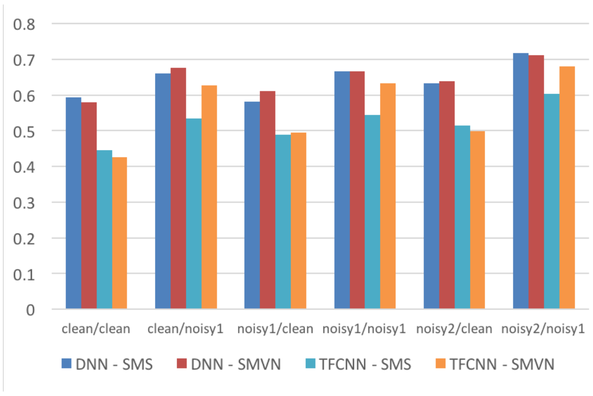

Figure 2: *$minC_{nxe}$ of various bottleneck architectures using SMS and SMVN normalizations on clean and renoised conditions from the QUESST 2014 task on T1 dev queries. Results obtained using DNN-CTS-SAD to trim the queries.*

It is also apparent that for all enrollment/test conditions, using SMS for feature normalization performed reliably better than SMVN. This seems especially true for the TFCNN. The only two exceptions to this observation are the clean/clean and noisy2/clean conditions, where SMVN offered a slight advantage over SMS.

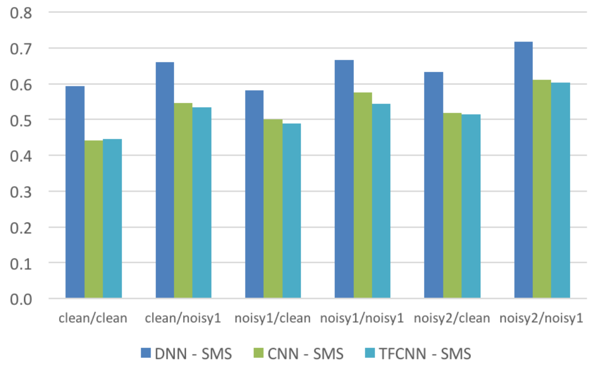

Figure 3: *$minC_{nxe}$ of various bottleneck architectures on clean and renoised conditions from the QUESST 2014 task on T1 dev queries. Results obtained using DNN-CTS-SAD to trim the queries.*

In Figure 3, we compare the DNN, CNN, and TFCNN architectures for bottleneck extraction and their performance on the QbE evaluation task. We only show the experiments using SMS, as this normalization performed better or equally well for all networks compared to the SMVN normalization. While in matched clean conditions, the CNN and TFCNN performed equally well, we see that in matched and mismatched noisy conditions the TFCNN offered a small yet reliable advantage over the CNN architecture. These gains ranged from 0.84% relative for the noisy2/clean condition, to 5.45% for the matched noisy1/noisy1 condition. The gains over the standard DNN architecture were of 18.77% on average and were quite consistent across all conditions.

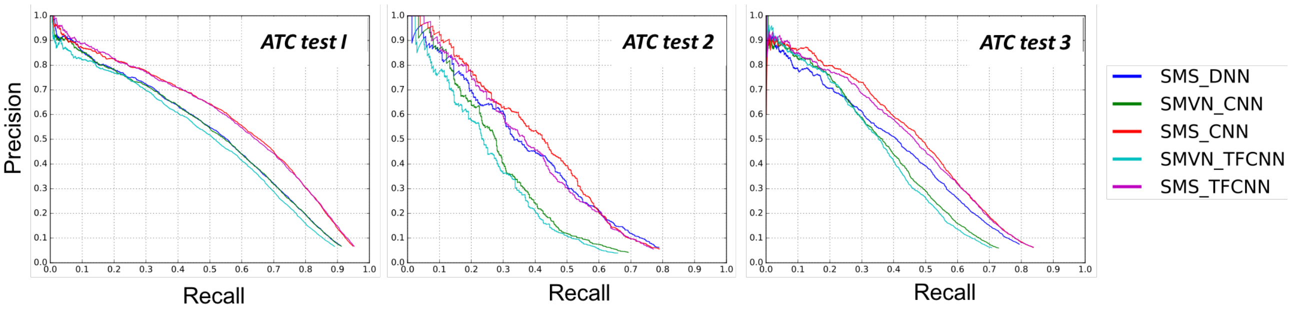

Figure 4: *QbE performance of various bottleneck architectures and normalization techniques on the ATC dataset. Results obtained using DNN-CTS-SAD to trim the queries. The three different test conditions refer to the enrollment and test sets defined in Section 4.2.*

Also in Figure 3, we observe that added noise reliably degraded performance, especially less stationary noises as were added in the $noisy2$ condition. Also, as a reminder, in none of these cases where the bottleneck networks retrained on those noises, as those were only applied on the enrollment and test data from MediaEval. It is therefore reassuring to see that the proposed CNN and TFCNN bottleneck architectures performed well above the standard DNN in cases of true unseen acoustic conditions.

On the Air Traffic Control dataset, we show results in the form of a Precision/Recall curve, which allows to look at various operating points at once while being more readable than other representations such as traditional DET curves. The performance of different neural network architectures and feature normalization are shown in Figure 4.

We first find that SMS normalization consistently outperforms SMVN normalization by a large margin and over all operating points. This finding confirms the preliminary conclusions on QUESST14 established earlier in this section. Then, we find that the CNN and TFCNN systems with SMS offer large gains in performance over the traditional DNN SMS bottleneck system, thus confirming that convolutional layers provide a considerable advantage when training bottleneck networks. Interestingly, we find that while the TFCNN and CNN perform very similarly in test condition 1, the CNN seems to consistently outperform the TFCNN by a small yet consistent margin when moving to test cases 2 and 3. The latter conditions introduced some large acoustic distortions due to speech transmitted from the planes, which is a true mismatch to the training data used to train the bottlenecks. Contrary to our findings on the augmented noise and reverb data QUESST14 data, it looks like the TFCNN doesn't generalize as well as the CNN on this type of unseen distortions. This adds to previous findings [10] where the TFCNN was found to perform particularly well on seen and unseen noisy and reverb data, but had never been tested on unseen, real radio transmissions. An interesting follow-up experiment to see if the TFCNN can learn the range of distortions seen in radio would be to retrain the bottleneck systems by adding similar radio data during training.

5.2. Impact of speech activity detection

In Table 2 we evaluate the impact of SAD on the QUESST14 T1 task. We observe a consistent gain from using SAD over not using SAD, and out of the two SAD compared, GP-HU-SAD obtains the best performance. As a reminder, both SADs have had their thresholds optimized independently. For the GP-HU-SAD the optimal threshold on the sum of speech posteriors was found to be 0.97, therefore any frame with score less than 0.97 was discarded. This is quite a high bar and probably a consequence of the relatively clean conditions of the data, and the types of artifacts (presence of telephone 'clicks' and pauses) which the SAD was trying to remove. The reason why the phonetic recognition-based SAD outperforms the DNN-based SAD is unclear, but we have a few hypotheses. It could be that the data used in training the GP-HU-SAD system is a closer match to the QUESST14 data. It could also be that because the GP-HU-SAD system stems from a phonetic recognizer where precise detection boundaries are important, it is giving more precise start and end times for speech and non-speech when used as a SAD. Conversely, the DNN-based SAD was developed to find coarse speech regions and is typically evaluated with a significant tolerance around the speech/non-speech boundaries. While we didn't apply any smoothing to the posteriors, which we found worked best for QbE, this may not be enough since the system itself was trained on annotations which didn't include precise boundaries.

As a comparison point with other state-of-art systems, our result of 37.1% in $minC_{nxe}$ can be directly compared to the 42.5% obtained by the winning group of the MediaEval QUESST 2014 evaluation on their best single-system "GP CZ BN", as can be seen in Table 2 of their paper [6]. Their system also uses the GP-HU-SAD and is very similar to the proposed in architecture, but instead of our multilingual CNN bottleneck used a mono-lingual stacked bottleneck trained with 20 hrs. of Czech-only data. It is worth noting that the authors of [6] report a $minC_{nxe}$ of 37.6% once this single system is calibrated using query-utterance and language-id side information, something that wasn't implemented in the proposed system to avoid tuning to a specific database.

Table 2: *Impact of SAD on the TFCNN-SMS QbE system on the QUESST2014 dev set using T1 queries. Comparison with other state-of-the-art systems.*

System	$minC_{nxe}$	maxTWV
no SAD	0.514	0.566
DNN-CTS-SAD	0.421	0.611
GP-HU-SAD	0.371	0.665
GP CZ BN [6]	0.425	-
GP CZ BN + sideinfo [6]	0.376	-

In Table 3 we look at the impact of SAD on QbE performance on the shorter and more distorted queries found in the ATC dataset. We apply both SADs over a wide range of thresh-

Table 3: *Impact of SAD on the CNN-SMS QbE system on the ATC dataset. The three different test conditions refer to enrollment and test data types, as defined in Section 4.2. The metric is Precision at two different Recall rates: 30% and 70%.*

SAD	Threshold	**ATC Test 1**		**ATC Test 2**		**ATC Test 3**	
		P@R=30%	P@R=70%	P@R=30%	P@R=70%	P@R=30%	P@R=70%
no SAD	n/a	**0.80**	**0.50**	**0.70**	**0.21**	**0.73**	**0.23**
DNN-CTS-SAD	1	0.79	**0.50**	0.69	0.18	0.72	0.22
	2	0.79	0.49	0.68	0.18	0.71	**0.23**
	2.5	0.79	0.49	0.68	0.16	0.71	**0.23**
	3	0.78	0.48	**0.70**	0.16	0.70	0.22
	3.5	0.77	0.45	0.63	0.15	0.66	0.20
GP-HU-SAD	0.95	0.68	0.29	0.47	n/a	0.50	0.09
	0.9	0.71	0.35	0.48	0.07	0.54	0.11
	0.7	0.76	0.41	0.57	0.11	0.64	0.16
	0.5	0.79	0.45	0.59	0.12	0.69	0.20
	0.3	0.79	0.46	0.64	0.14	0.69	0.20
	0.1	0.79	0.48	0.68	0.16	0.72	0.21

olds and find that no threshold provides a consistently better level of performance over using no SAD at all. It is worth noting that while the GP-HU-SAD was trained on clean telephone only, the DNN-CTS-SAD was trained on telephone speech retransmitted radio channels meant to simulate a subset of radio distortions but was still unable to consistently outperform using no SAD. This is likely due to the fast rate of speech found in this corpus, as well as the precise boundaries provided by queries extracted from within a sentence.

Future work will look at how augmentation and adaptation can help improve performance of bottleneck networks in specific target conditions with limited to no annotated in-domain data. The authors would also like to compare and combine the proposed convolutional bottleneck network architectures with more complex two-step hierarchical bottleneck architectures, such those as described in [4, 5].

6. Conclusions

In this paper, we presented two simple bottleneck architectures previously used in ASR and showed how they are applicable to DTW-based QbE in unseen and challenging acoustic conditions. We evaluated the performance of the proposed architecture on the MediaEval QUESST 2014 QbE task as well as on Air Traffic Control data, and showed that these architectures outperformed standard DNN bottlenecks over a variety of matched and mismatched enrollment and test conditions. We also showed that in some unseen noisy and reverberated conditions, the time convolution in the TFCNN architecture can provide some gain over the standard CNN architecture, but that those gains don't hold on unseen radio data such as seen in the ATC testbed. Finally, we provided insights over feature-level normalization and showed that spectral mean subtraction is more robust than mean-variance normalization over the range of test conditions.

7. References

[1] J. van Hout, V. Mitra, Y. Lei, D. Vergyri, M. Graciarena, A. Mandal, and H. Franco, "Recent improvements in SRI's keyword detection system for noisy audio," in *Proc. Interspeech 2014, Singapore*, 2014.

[2] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *IEEE Spoken Language Technology Workshop (SLT)*, Dec 2012, pp. 336–341.

[3] J. Hou, V. T. Pham, C.-C. Leung, L. Wang, H. Xu, H. Lv, L. Xie, Z. Fu, C. Ni, X. Xiao, H. Chen, S. Zhang, S. Sun, Y. Yuan, P. Li, T. L. Nwe, S. Sivadas, B. Ma, C. E. Siong, and H. Li, "The NNI Query-by-Example system for MediaEval 2015," in *MediaEval*, 2015.

[4] F. Grézl and M. Karafiát, "Hierarchical neural net architectures for feature extraction in ASR," in *INTERSPEECH*, 2010.

[5] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, Dec 2011, pp. 42–47.

[6] I. Szöke, M. Skácel, L. Burget, and J. Černockỳ, "Coping-with channel mismatch in query-by-example-but quesst 2014," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5838–5842.

[7] T. N. Sainath, A. r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8614–8618.

[8] O. Abdel-Hamid, A. r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 4277–4280.

[9] V. Mitra, W. Wang, H. Franco, Y. Lei, C. Bartels, and M. Graciarena, "Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions," in *Proc. Interspeech 2014, Singapore*, 2014.

[10] V. Mitra and H. Franco, "Time-frequency convolutional networks for robust speech recognition," in *Proc. 2015 IEEE Automatic Speech Recognition and Understanding Workshop*, 2015.

[11] S. Ganapathy, K. Han, S. Thomas, M. Omar, M. V. Segbroeck, and S. S. Narayanan, "Robust language identification using convolutional neural network feature," in *Proc. Interspeech 2014, Singapore*, 2014.

[12] C. Bartels, W. Wang, V. Mitra, C. Richey, A. Kathol, D. Vergyri, H. Bratt, and C. Hung, "Toward human-assisted lexical unit discovery without text resources," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 64–70.

[13] J. Qi, D. Wang, J. Xu, and J. Tejedor, "Bottleneck features based on gammatone frequency cepstral coefficients," in *Proc. Interspeech*, F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, Eds. ISCA, 2013, pp. 1751–1755.

[14] M. Muller, *Information Retrieval for Music and Motion*. Springer-Verlag, 2007.

[15] X. Anguera and M. Ferrarons, “Memory efficient subsequence DTW for query-by-example spoken term detection,” in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, July 2013, pp. 1–6.

[16] 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.

[17] J. Godfrey, “Air Traffic Control Complete LDC94S14A,” 1994, web Download. Philadelphia: Linguistic Data Consortium.