Query-By-Example Spoken Term Detection Using Generative Adversarial Network

Neil Shah*§, Sreeraj R§, Maulik C Madhavi‡, Nirmesh J. Shah§ and Hemant A. Patil§
§ Dhirubahi Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, India
E-mail: {neilshahnms, srpillai88}@gmail.com, {nirmesh88_shah, hemant_patil}@daiict.ac.in
* TCS Research, Tata Consultancy Services Pvt. Ltd., Pune, India
‡ Department of Electrical and Computer Engineering, National University of Singapore, Singapore
E-mail: maulik.madhavi@nus.edu.sg

***Abstract*—Several Neural Network (NN)-based representation techniques have already been proposed for Query-by-Example Spoken Term Detection (QbE-STD) task. The recent advancement in Generative Adversarial Network (GAN) for several speech technology applications, motivated us to explore the GAN in QbE-STD. In this work, we propose to exploit GAN with the regularized cross-entropy loss, and develop a framework featuring GAN, trained using Gaussian Mixture Model (GMM)-based posterior labels. The proposed GAN maps the speech-specific features to the unsupervised posterior labels. This mapping represents the speech through an unsupervised GAN posteriorgram (uGAN-PG), for matching the query (keyword) with the utterances in the document. The QbE-STD, using the proposed posteriorgram is performed on the TIMIT database. We compare the performance of the proposed uGAN-PG with the unsupervised Deep Neural Network (DNN) posteriorgram (uDNN-PG) and obtained the relative performance improvement of 10.32 % Mean Average Precision and 5.6 % Precision by considering top N queries (p@N) over uDNN-PG.**

Index Terms: Query-by-Example, Generative Adversarial Network, Posteriorgram, Spoken term detection

I. Introduction

Query-by-Example Spoken Term Detection (QbE-STD) is the process of retrieving relevant documents from the entire speech corpus, using an audio query (i.e., keyword is in spoken audio form) [1]–[4]. The QbE-STD follows acoustic signal-level matching of speech documents with the audio query instead of matching the transcribed audio. With an increase in the availability of online audio that contains multi-languages and Out-of-Vocabulary (OOV) words, transcribing entire speech collection cannot offer a generic solution to the problem of audio search [5], [6]. Hence, improving the performance of QbE-STD using recent and efficient techniques is of utmost importance. MediaEval Spoken Web Search 2011 task was initiated to retrieve language-independent spoken content for low resource languages [1], [5], [7].

One of the main challenges in QbE-STD is to obtain a speaker-independent representation of audio signals for matching and precise retrieval [8]–[10]. Among the different proposed representations, unsupervised Gaussian Posteriorgrams and its recent variants, such as Dirchlet Process Gaussian Mixture Model (DPGMM), have been explored in QbE-STD task [11], [12]. Furthermore, Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM) [13], Unsupervised Bottleneck Features (uBNF) [14], Vocal Tract Length Normalized (VTLN) [15]–[18] and Unsupervised Deep Neural Network-based posteriorgrams (uDNN-PG) [14] have been proposed to provide a better representation for the speech. uBNF and uDNN-PG use labeled posteriorgram as in supervision to obtain better posteriorgrams, while the Gaussian Posteriorgram (GP) is unsupervised [14].

The posteriorgram-based methods, such as GMM, DNN, etc. uses Maximum Likelihood (ML)-based optimization, that assumes the output variables follows Gaussian distribution. This prior assumption on the data distribution may prevent the network from its optimization [19]–[21]. However, Generative Adversarial Network (GAN) adversarially optimizes their parameters by not posing any specific prior assumption on the data. Hence, in this paper, we propose to exploit GAN as an alternative to the DNN-based posterior feature representation for the QbE-STD. We present an unsupervised GAN-based posterior representation (uGAN-PG), which is trained on unsupervised labeled GMM posteriorgram [14]. The proposed regularized adversarial network, uGAN performs better than uDNN, during the objective evaluation on a few selected feature sets. The proposed system is evaluated using Mean Average Precision (MAP) and precision at N (p@N) [22]. To the best of authors' knowledge, this is the first study, which proposes to exploit the potential of GAN for the QbE-STD task.

II. Unsupervised Feature Representation

A. Unsupervised Posterior Label

Alike [14], in this work, we consider unsupervised GMM to represent the speech in a predetermined number of clusters. For *N*-component GMM, the posteriorgram for a speech frame, x_k, from entire M frames, $X = \{x_i\}_{i=1}^{M}$, is given by $p_k = (p_{k,1}, p_{k,2}, ..., p_{k,N})$, where $p_{k,n}$ denotes the posterior probability of the k^{th} speech frame to be generated by the n^{th} Gaussian component [11], [23]. We follow the same procedure in labeling the posteriorgram as suggested in [14], [24]. For a frame x_i, the label l_i is a vector of N elements, similar to the posteriorgram with its maximum posterior probability component set to '1', and value of all the other components set to '0'. l_i is referred as the labelled posteriorgram for x_i.

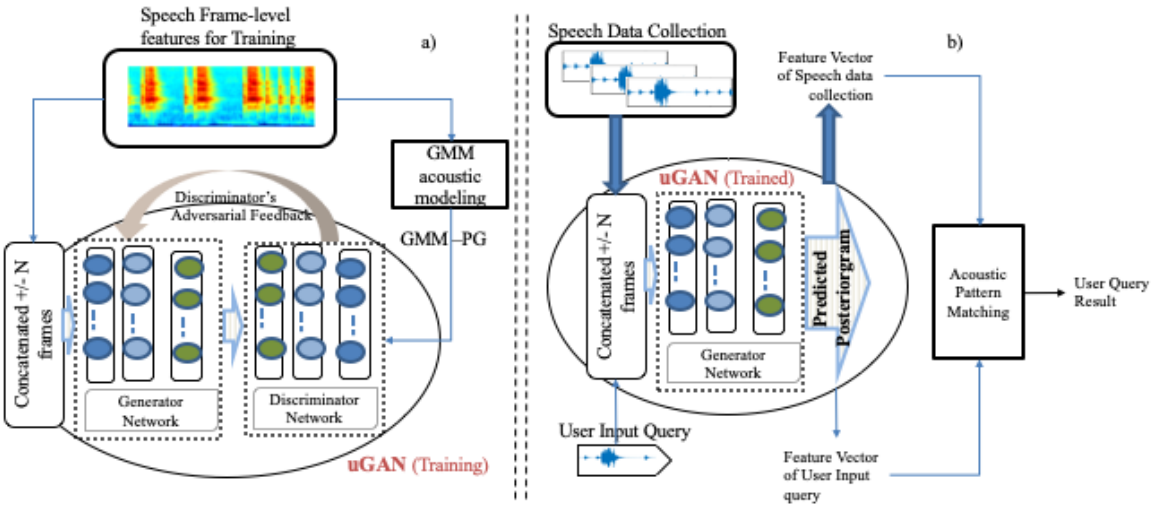

Fig. 1. Proposed framework of QbE-STD based on unsupervised GAN-based posteriorgram features (uGANs). (a) trains the uGAN for generating posteriorgram which then can be used for acoustic pattern matching with the user input query as shown in (b).

B. Unsupervised GAN Posteriorgram (uGAN-PG)

GAN is a generative model that learns deep representation through the adversarial training between a pair of networks in competition with each other [19], [25]. GAN produces the samples, resembling the data distribution $\mathcal{X}$, by learning the mapping function between the samples y from some prior distribution $\mathcal{Y}$ to the samples x belonging to $\mathcal{X}$. Here, we propose to exploit these characteristics in learning the posterior-like representation of the audio queries. Here, the objective of the generator (G) is to generate posterior-like representation and the objective of the discriminator (D) is to differentiate between the labeled-GMM posteriorgram and the output generated by the G network.

Regularization of the adversarial objective function facilitates the network in learning the desired representation corresponding to the given input, which otherwise may fail [12], [20], [21], [26]–[28]. Optimization through cross-entropy (CE) loss function helps the network to reduce the divergence between the data distribution and the predicted distribution [29]. The regularized G network objective function (i.e. $V(G)$)) with the cross-entropy loss can be given by:

$$\min_G V(G) = -\mathbb{E}_{y\sim\mathcal{Y}}[\log(D(G(y)))] + \mathbb{E}_{x\sim\mathcal{X},y\sim\mathcal{Y}}[x\log G(y) + (1-x)\log(1-G(y))]. \quad (1)$$

Training a DNN for unsupervised clustering of data is challenging [30]. However, as suggested in [31], a weak classifier output can be used as labels for training a strong classifier network. This provides feasibility to use DNN in the low resource problem. It has been established in [14] that the uDNN trained on GMM-based labels mostly outperforms the conventional GMM approach. In this paper, we compare the performance of uDNN model, with the proposed uGAN model. We use GMM-based labels for training both uDNN and uGAN in the QbE-STD, as suggested in [14].

III. Details of QbE-STD

The proposed QbE-STD framework is exemplified in Fig. 1. The proposed uGAN-PGs and uDNN-PGs are extracted by introducing cepstral frame-level features to the uGAN and uDNN, respectively. This is followed by taking the output from the last layer of the G network of uGAN and uDNN. In this study, we use subsequence-DTW (SDTW) along with symmetric Kullback Leibler (KL) divergence for distance computation and pattern matching [32], [33]. Let 'q' and 'u' be the N component frame-level posteriorgram representation of query and utterance [2]. The symmetric KL distance between q and u is given by:

$$d_{q,u} = \sum_{i=1}^{N}\left[q(i)\log\left(\frac{q(i)}{u(i)}\right) + u(i)\log\left(\frac{u(i)}{q(i)}\right)\right]. \quad (2)$$

SDTW is applied to the obtained distance matrix to fetch the optimal warping path that detects the matching query

pattern in an utterance, with a minimum distance cost function. The cost value is equivalent to the dissimilarity between the frames. Based on the cost calculated for all the utterances in the speech data collection, the most relevant documents are retrieved. Traditional evaluation measures, such as mean Average Precision (MAP), precision at N (p@N) are used as defined in [2], [22].

IV. Experimental Setup

A. Dataset Used

The proposed system is evaluated on the TIMIT database [34]. The training utterances are divided into a training set and validation set with a ratio of 9:1 [14]. For testing, we have extracted 503 queries, with 84 unique queries from speakers that are disjoint from the training set. Average precision is calculated over all the unique queries and then MAP is computed for the entire dataset. Experiments are conducted with, *36*-dimension (d) Mel filterbank (FBank) as Feat1, *72*-d FBank+Δ (*36*-d FBank + *36*-d Δ) as Feat2 and 39-d MFCC+Δ+$\Delta\Delta$ (13 each) as Feat3. All the feature sets are extracted using a 25 ms window, with a 10 ms shift, and are post-processed by the mean and variance normalization [2].

B. Network Setup

The uDNN and uGAN are trained for all the aforementioned feature sets along with the varying GMM components and the context size. Initially, GMM is trained with the *39*-d MFCC, which acts as the baseline weak classifier feature set [14]. A different number of GMM components are extracted from 64, 128, and 256 in the baseline GP system, and the GP is used as a label (target) for uDNN and uGAN systems. While the [+/-I] contextual features are passed to the input side, the labeled central frame posteriorgram is fed to the output layer of the network for training. In uGAN, the G network is kept identical to uDNN, for analyzing the performance of adversarial optimization over the ML-based optimization [21]. The uDNN and G network has four hidden layers. Each layer has 1024 hidden units, followed by batch normalization and nonlinear activation (as used in [35]). A significant improvement in the performance is noted by the inclusion of the batch normalization [36], before applying the nonlinear activation and the dropout [37]. The output layer has (64,128,256) units, depending on the number of components, respectively, for all the three feature sets. At each layer, the dropout rate of 0.5 is maintained [37]. The output layer uses softmax operation, to produce posteriorgram-equivalent feature representation, while all the other layers use sigmoid activation. However, we use labeled posteriorgram (1-hot) vector representation to use CE loss during training.

TABLE I
Configurations of GANs and DNNs

Inputs	Ndim	Context	DNN and G	D
Feat1	36	+/-1,+/-3,+/-5	1024x4,64/128/256	512x3,64
Feat2	72	+/-1,+/-3,+/-5	1024x4,64/128/256	512x3,64
Feat3	39	+/-1,+/-3,+/-5	1024x4,64/128/256	512x3,64

The D network of uGAN has three hidden layers, with 512 hidden units in each layer. The D network also employs batch normalization, followed by the *tanh* nonlinear activation. The output layer has 64 units with sigmoid activation. An Adam optimizer [38] is used for minimizing the cross-entropy, with a learning rate of 0.001. All the networks are trained for 150 epochs, with an effective batch size of 1000. The details of this configuration are shown in Table I.

V. Experimental Results

Fig. 2 compares the posteriorgram for the query '*intelligence*' for uDNN and uGAN models. The red dash-dot circle represents the ambiguity in posterior probability across the components in uDNN, which may lead to false detection. However, the probabilities are more compactly represented by the proposed uGAN-PGs.

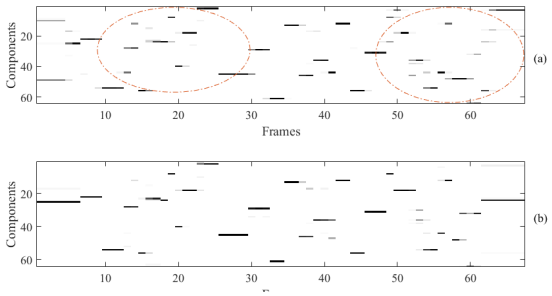

Fig. 2. Posteriorgram representation for the query '*Intelligence*' for (a) uDNN, and (b) uGAN. More ambiguity in uDNN-PG can be inferred from the regions shown via dotted circles.

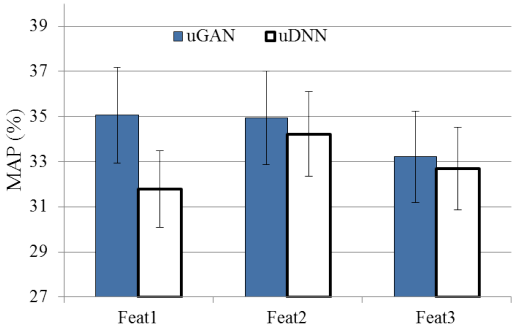

Fig. 3. MAP performance for uGAN-PG and uDNN-PG, along with their 95 % confidence interval, for all the three feature sets, with a specific 256 GMM components and +/- 1 context length.

The feature-level analysis on both the systems from Fig. 3, shows the better posterior representation captured by the FBank features (notably *36*-d FBank) over MFCC, as also analyzed in [39]. Moreover, for all the feature sets and with the selected configuration, uGAN significantly outperforms the uDNN. Table II shows the performance comparison between the proposed uGAN-PGs and uDNN-PGs, for all the three feature sets, with varying GMM components and context size.

TABLE II
PERFORMANCE COMPARISONS AND THE 95 % CONFIDENCE INTERVAL COMPUTED FOR UGAN-PG AND UDNN-PG FOR ALL THE FEATURE SETS, DIFFERENT NUMBER OF GMM COMPONENTS AND CONTEXT SIZE

Feature set	*N*	+/-5 GAN (uGAN-PG)	+/-5 DNN (uDNN-PG)	+/-3 GAN (uGAN-PG)	+/-3 DNN (uDNN-PG)	+/-1 GAN (uGAN-PG)	+/-1 DNN (uDNN-PG)
Feat1	**64**	**27.25±2.11**	25.87±1.84	29.33±2.05	31.65±1.84	28.20±2.18	30.78±1.75
		(26.07±4.79)	(25.95±4.25)	(26.48±4.70)	(30.31±4.02)	(26.45±5.12)	(30.13±3.96)
(*36*-d)	**128**	30.54±2.08	30.56±1.85	32.20±2.05	34.25±1.92	29.02±2.00	31.36±1.73
		(27.83±4.73)	(30.53±4.25)	(30.52±4.53)	(32.67±4.24)	(27.71±4.52)	(30.90±3.68)
	256	31.85±2.06	32.14±1.82	**35.15±2.08**	34.92±1.84	**35.06±2.11**	31.78±1.70
		(30.72±4.56)	(31.60±4.17)	(33.81±4.49)	(33.73±4.18)	(33.48±4.69)	(31.70±3.81)
Feat2	**64**	**27.37±2.08**	27.31±1.88	**29.42±2.06**	28.62±1.89	29.47±2.08	30.58±2.01
		(26.29±4.94)	(26.94±4.33)	(26.969±4.74)	(28.78±4.40)	(27.10±4.74)	(28.42±4.60)
(*72*-d)	**128**	30.05±2.07	31.26±1.91	**32.27±2.01**	31.97±1.92	**32.59±2.09**	32.54±1.91
		(28.63±4.90)	(30.61±4.28)	(30.12±4.49)	(30.60±4.22)	(31.06±4.60)	(31.70±4.39)
	256	31.98±2.06	32.08±1.84	**34.90±2.07**	33.21±1.88	**34.93±2.07**	34.22±1.87
		(31.05±4.61)	(31.71±4.09)	(33.49±4.47)	(32.59±4.28)	(32.91±4.53)	(32.86±3.96)
Feat3	**64**	**25.91±1.95**	25.81±1.76	**28.39±1.95**	27.48±1.78	27.97±1.97	28.86±1.82
		(23.72±4.71)	(25.81±4.10)	(26.64±4.62)	(26.56±4.14)	(26.40±4.54)	(27.52±4.39)
(*39*-d)	**128**	29.19±2.03	30.97±1.87	31.46±2.02	32.10±1.78	31.38±2.02	31.82±1.76
		(27.70±4.68)	(30.75±4.27)	(29.46±4.69)	(30.97±4.01)	(29.80±4.61)	(30.57±4.04)
	256	31.17±2.04	31.22±1.78	**33.01±2.04**	32.48±1.84	**33.22±2.02**	32.69±1.84
		(29.30±4.57)	(30.35±4.05)	(31.04±4.39)	(31.68±4.15)	(30.97±4.61)	(32.01±4.24)

N indicates the number of GMM components, +/-I indicates I frames to the left and right of the central frame, '±' indicates margin of error corresponding to 95 % CI, the bold content indicates the case, where uGAN outperforms uDNN, and figures in round brackets shows the p@N.

TABLE III
AN ANALYSIS ON RELATIVE PERFORMANCE (IN %) OF UGAN-PG OVER UDNN-PG FOR *36*-D FBANK FEATURES

N	+/-5	+/-3	+/- 1
64	**5.33 %** ↑	7.9 % ↓	9.14 % ↓
128	0.6 % ↓	6.3 % ↓	8.06 % ↓
256	0.9 % ↓	**0.66 %** ↑	**10.32 %** ↑

↑ shows performance increment and ↓ shows performance decrement.

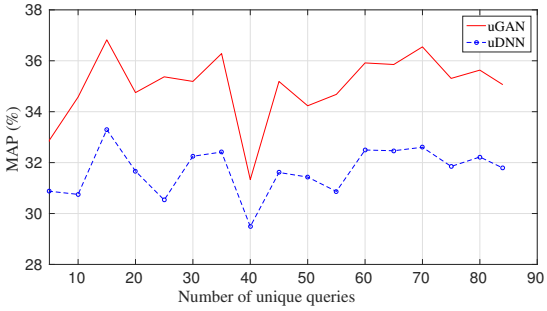

Fig. 4. Performance comparision with *36*-d FBank features, with 256 GMM components and +/-1 context length.

Table III shows the relative improvement (bold figures) gained by uGAN-PG for *36*-d FBank features over uDNN-PG. The posteriorgram represented by uGAN with +/- 1 context size and 256 GMM components, results in 10.32 % relative increment in MAP over uDNN-PG. On the other hand, the uGAN-PG with +/- 5 context length and 64 GMM components, results in 5.33 % of relative improvement in MAP. These dictate the trade-off in selecting the context length and GMM labels, while training uGAN. The total number of English phoneme set in TIMIT is set to 61. However, the interspeaker and intraspeaker variations in the acoustic pattern can be captured more effectively by increasing the number of components, which can be inferred from Table II, for all the feature sets. On the contrary, when the context length is increased, the number of parameters to be learned increases, which leads to overfitting. However, when the Gaussian components are further increased to 512, the relative decrement of 11.4 % MAP and 9.8 % p@N is observed for uGAN-PG.

The performance comparison between uGAN and uDNN (*36*-D FBank, 256 GMM components, and +/- 1 context length) shown in Fig. 4 displays the average MAP scores obtained for random selection of 84 unique queries repeated 10 times for each set. Each set contains an increment of 5 random unique queries. The margin of error corresponding to the 95 % CI for uGAN-PG and uDNN-PG is 2.11 and 1.70, respectively. The high range of MAP in uGAN over uDNN, as visible from Fig. 4 accounts for the higher error margin in uGAN. Moreover, the p@N analysis on the same feature set demonstrates the relative improvement of 5.61 % by the uGAN-PGs. This shows the potential of adversarial optimization as a statistically significant alternative to an ML-based optimization for the QbE-STD task.

VI. CONCLUSIONS

In this work, we proposed a framework for QbE-STD using a Generative Adversarial Network (GAN). In particular, a DNN-based GAN with a cross-entropy regularization is employed for extracting an unsupervised posterior feature representation (uGAN-PG), trained on labeled GMM posteriorgram. The uGAN-PG extracted using *36*-D Mel FBank features, with 256 GMM components and +/- 1 context length, generates the posteriorgram with 10.32 % and 5.61 % relative improvement in MAP and p@N, respectively, over uDNN-PG. Moreover, the statistical analysis on the selected feature set reveals the significant improvement in precision, irrespective of any randomly selected set of queries. Our future work

will include exploring the recent advancements in the GAN architecture (such as Wassertian GAN) for the QbE-STD task. Moreover, the GAN-based adversarial framework can be regularized with energy-based regularizer which is known to reduce the redundancy and induces generalization.

VII. Acknowledgements

We sincerely thank authorities of DA-IICT, Gandhinagar, TCS Media Entertainment & Advertising Research, Pune and NUS Singapore for their kind support.

References

[1] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *IEEE ASRU*, Merano, Italy, 2009, pp. 421–426.

[2] M. Madhavi, "Design of QbE-STD-system: Audio representation and matching perspective," Ph.D. Thesis, Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, India, Nov, 2017.

[3] M. C. Madhavi and H. A. Patil, "Design of mixture of GMMs for query-by-example spoken term detection," *Computer Speech & Language*, vol. 52, pp. 41–55, 2018.

[4] M. Madhavi and H. A. Patil, "Combining evidences from detection sources for query-by-example spoken term detection," in *APSIPA-ASC*, Kuala Lumpur, Malaysia, 2017.

[5] A. Mandal, K. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: A survey," *IJST, Springer*, vol. 17, no. 2, pp. 183–198, 2014.

[6] B. Elizalde, S. Zarar, and B. Raj, "Cross modal audio search and retrieval with joint embeddings based on text and audio," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 4095–4099.

[7] X. Anguera, L. J. Rodríguez-Fuentes, I. Szöke, A. Buzo, F. Metze, and M. Peñagarikano, "Query-by-example spoken term detection evaluation on low-resource languages," in *SLTU*, St. Petersburg, Russia, 2014, pp. 24–31.

[8] A. Saxena and B. Yegnanarayana, "Distinctive feature based representation of speech for query-by-example spoken term detection," in *INTERSPEECH*, Dresden, Germany, 2015, pp. 3680–3684.

[9] A. Asaei *et al.*, "Phonological posterior hashing for query by example spoken term detection," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 2067–2071.

[10] Z. Zhu, Z. Wu, R. Li, H. Meng, and L. Cai, "Siamese recurrent auto-encoder representation for query-by-example spoken term detection," *INTERSPEECH*, pp. 102–106, 2018.

[11] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *IEEE ASRU*, Merano, Italy, 2009, pp. 398–403.

[12] H. Chen, C. Leung, L. Xie, B. Ma, and H. Li, "Parallel inference of Dirichlet Process Gaussian Mixture Models for unsupervised acoustic modeling: A feasibility study," in *INTERSPEECH*, Dresden, Germany, 2015, pp. 3189–3193.

[13] R. R. Pappagari, S. Nayak, and K. S. R. Murty, "Unsupervised spoken word retrieval using Gaussian-Bernoulli restricted Boltzmann machines," in *INTERSPEECH*, Singapore, 2014, pp. 1737–1741.

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

[15] N. J. Shah *et al.*, "Unsupervised vocal tract length warped posterior features for non-parallel voice conversion," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 1968–1972.

[16] N. J. Shah, R. Sreeraj, N. Shah, and H. A. Patil, "Novel inter mixture weighted GMM posteriorgram for DNN and GAN-based voice conversion," in *APSIPA-ASC*, Hawaii, USA, 2018, pp. 1776–1781.

[17] M. C. Madhavi, S. Sharma, and H. A. Patil, "Vocal tract length normalization features for audio search," in *International Conference on Text, Speech and Dialogue (TSD)*, Pilsen, Czech Republic, 2015, pp. 387–395.

[18] M. C. Madhavi and H. A. Patil, "VTLN-warped Gaussian posteriorgram for QbE-STD," in *EUSIPCO*, Kos Island, Greece, 2017, pp. 563–567.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, Montreal, Canada, 2014, pp. 2672–2680.

[20] N. Shah, H. A. Patil, and M. H. Soni, "Time-frequency mask-based speech enhancement using convolutional generative adversarial network," in *APSIPA-ASC*, Hawaii, USA, 2018, pp. 1246–1251.

[21] M. H. Soni, N. Shah, and H. A. Patil, "Time-frequency masking-based speech enhancement using generative adversarial network," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018.

[22] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," 2009, http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf (Last Accessed on Oct. 11, 2020).

[23] M. C. Madhavi and H. A. Patil, "Partial matching and search space reduction for QbE-STD," *Comput. Speech Lang.*, vol. 45, no. 3, pp. 58–82, 2017.

[24] H. Chen, C. Leung, L. Xie, B. Ma, and H. Li, "Multitask feature learning for low-resource query-by-example spoken term detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1329–1339, 2017.

[25] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[26] D. Michelsanti and Z.-H. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3642–3646.

[27] N. Shah, N. J. Shah, and H. A. Patil, "Effectiveness of generative adversarial network for non-audible murmur-to-whisper speech conversion." in *INTERSPEECH*, 2018, pp. 3157–3161.

[28] N. J. Shah, M. Parmar, N. Shah, and H. A. Patil, "Novel MMSE DiscoGAN for cross-domain whisper-to-speech conversion," in *Machine Learning in Speech and Language Processing (MLSLP) Workshop*, Google Office, Hyderabad, India, 2018, pp. 1–3.

[29] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[30] D. Ram, A. Asaei, and H. Bourlard, "Subspace detection of DNN posterior probabilities via sparse representation for query by example spoken term detection," in *INTERSPEECH*, San Francisco, USA, 2016, pp. 918–922.

[31] Y. Zhang, R. Salakhutdinov, H. A. Chang, and J. Glass, "Resource configurable spoken query detection using deep boltzmann machines," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 5161–5164.

[32] Y. Zhang and J. R. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 5660–5663.

[33] A. Muscariello, G. Gravier, and F. Bimbot, "Audio keyword extraction by unsupervised word discovery," in *INTERSPEECH*, Brighton, United Kingdom, 2009, pp. 2843–2846.

[34] J. S. Garofolo, "DARPA TIMIT acoustic-phonetic speech database," *NIST*, vol. 15, pp. 29–50, 1988.

[35] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 1993–1997.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on MAchine Learning (ICML)*, Lille, France, 2015, pp. 448–456.

[37] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[38] D. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *ICLR*, San Diego, USA, 2015, pp. 1–15.

[39] 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.