SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I7P110

Volume 12 Issue 7, 119-136, July 2025
© 2025 Seventh Sense Research Group®

Review Article

Query-by-Example Spoken Term Detection: A Systematic Review

Manisha Naik Gaonkar[1,2], Veena Thenakanidiyoor[1], A. D. Dileep[3]

[1]*Deparment of Computer Science and Engineering, NIT Goa, Goa, India.*
[2]*Department of Computer Engineering, Goa College of Engineering, Goa, India.*
[3]*Department of Computer Science and Engineering, IIT Dharwad, Karnataka, India.*

[1]*Corresponding Author : gaonkar.mani@gmail.com*

Received: 05 May 2025 Revised: 07 June 2025 Accepted: 08 July 2025 Published: 31 July 2025

Abstract - *Massive collections of multimedia data have been created as a result of the Internet's recent exponential expansion. Effective management of these repositories necessitates matching spoken queries with the audio content of these videos. One powerful technique that has emerged to address this issue is QbE-STD. Nevertheless, QbE-STD encounters many difficulties, including age sensitivity, dialect differences, and computational complexity. The objective of this study is to address these issues. In this area, review publications are scarce and mostly lack an in-depth study of feature representations and matching techniques. This paper presents a detailed analysis of different approaches and developments in QbE-STD. It covers feature representations, similarity metrics, matching methods, datasets, evaluation measures, and benchmarking platforms. The paper delves into the intricacies of various feature representations and scrutinizes similarity metrics. These metrics are analyzed for their advantages and disadvantages in computing a matching matrix between a query and an utterance. Furthermore, the paper highlights how machine learning and deep learning architectures are increasingly integrated into QbE-STD. Finally, the paper discusses a few challenges associated with QbE-STD, which provide an opportunity for future research in this field.*

Keywords - *Convolutional Neural Network, Spoken term detection, Query-by-Example Spoken Term Detection, Keyword spotting, Audio search.*

1. Introduction

The availability of high-speed networks at low cost has led to a huge increase in the generation of audio data. Audio data is generated through various sources: lecture recordings from Massive Open Online Courses (MOOCs), YouTube videos, news channels, and TV recordings. With vast amounts of spoken data available, searching these audio databases becomes necessary to locate the required data. Searching in large audio databases is a technically challenging task. This comprehensive review explores the most recent developments, methodologies, datasets, and challenges in QbE-STD. By examining the various components of QbE-STD, including feature extraction, template matching, similarity measures, databases, benchmarking platforms, and evaluation metrics, this review provides a thorough understanding of the techniques that drive the QbE-STD systems. This review provides a comprehensive overview of the current landscape and identifies challenges and potential avenues for future research.

Audio search refers to searching for a query or a keyword in a database of audio utterances. Audio search is very challenging as the audio data varies a lot. When the same speaker uses the same word in different contexts, the speech signal for that word could vary. The search becomes more difficult due to variations in age, dialect, and gender, and it is also computationally expensive. There are many applications for audio search, such as in music retrieval systems, information retrieval, consumer search, and indexing audio archives [1].

Depending on the type of query provided, audio search is classified into two broad categories: Keyword Spotting (KWS) and Spoken Term Detection (STD). In KWS, the keywords are predefined, and the search is restricted to those predefined keywords only [2]. The keywords and utterances are converted to text, and a text-based search is carried out [3]. STD refers to searching audio databases for any query, so the keywords are not predetermined.

STD can be categorized into two categories depending on how the query is provided and how the search is carried out. They are text-based STD [4] and query-by-example STD (QbE-STD) [5]. Text-based STD involves converting the query to text format and performing a search based on text

[6]. Text-based speech recognition (STD) uses text-based search techniques with an Automatic Speech Recognition (ASR) system after the speech is converted to text. The problem with this approach is the need for an effective ASR system. Hence, this approach may not be suitable for a language with limited resources. In QbE-STD, a query is in the form of spoken audio and is matched with the repository of audio utterances [7]. QbE-STD does not involve speech-to-text conversion. It enables us to search in a multilingual speech database without using any speech recognition system. Figure 1 illustrates the types of audio search.

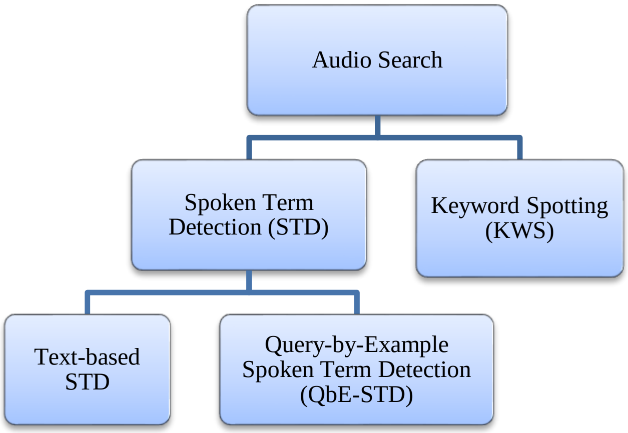
Audio Search
Spoken Term Detection (STD)
Keyword Spotting (KWS)
Text-based STD
Query-by-Example Spoken Term Detection (QbE-STD)

Fig. 1 Types of audio search

QbE-STD is regarded as a zero-resource system because no language-specific resources are used, and the search in QbE-STD is performed solely using the spoken query. It allows us to search various speech databases without an automatic speech recognition system by specifying a spoken query, making it an unsupervised pattern-matching problem. Also, the query and the audio utterance may be of different lengths. Therefore, the more significant audio signal must be broken down into smaller fragments that can be compared with the query.

QbE-STD is searching the audio query in spoken audio utterances. Figure 2 illustrates the general architecture of a QbE-STD system [5]. Feature representation refers to extracting the features from the audio query and utterances and suitably representing these features.

Template matching refers to the approaches used to match these feature representations of a query and an utterance to decide whether the query appears in the utterance or not. Approaches for QbE-STD mainly differ concerning feature representation and template matching. Different methodologies used for QbE-STD are analyzed in this paper.

Dynamic time warping (DTW) is the state-of-the-art technique used for QbE-STD, but it is computationally expensive [5]. Different feature representations, such as Gaussian posteriorgrams [9], bottleneck features [22], acoustic word embeddings [33], and transformer-based representations [35], have been used, but no single representation is universally optimal for QbE-STD.

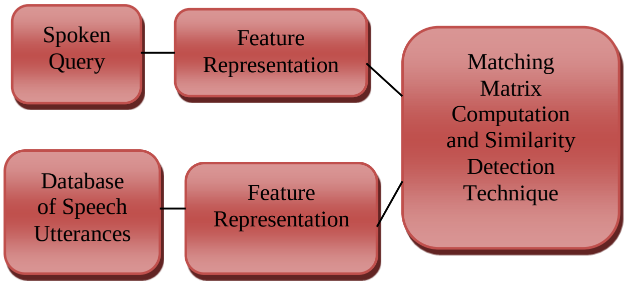
Spoken Query
Feature Representation
Database of Speech Utterances
Feature Representation
Matching Matrix Computation and Similarity Detection Technique

Fig. 2 A general architecture of the QbE-STD system

Further, various matching metrics such as Kullback-Leibler divergence [45], cosine similarity [5], and histogram intersection kernel [43] have been investigated to compute the matching matrix necessary for QbE-STD. Various deep learning strategies, such as CNN [43], Siamese network, and transformer-based architecture [36], have been explored to determine the relevance of the query to the reference utterance. The use of deep learning strategies has significantly improved the performance of the QbE-STD, especially in cross-lingual scenarios. Furthermore, techniques like parallel computation [82] and feature reduction [81] have been explored to improve computational efficiency. However, challenges such as scalability, noise robustness, and generalization to unseen conditions remain critical areas of research.

The rest of the paper is organized as follows: Section 2 discusses the applications of QbE-STD, Section 3 describes the methodology used for the review, Section 4 discusses the feature representations used for QbE-STD, while Section 5 explains the computation of the matching matrix. Section 6 discusses the various similarity metrics used for computing the matching matrix, and Section 7 discusses the different datasets available for QbE-STD, while Section 8 gives details of various evaluation metrics used for QbE-STD. In Section 9, benchmarking platforms are discussed. The challenges and future directions are provided in Section 10, a discussion in Section 11, and a conclusion of the paper in Section 12.

2. Applications of QbE-STD

QbE-STD is used in numerous applications. Some of the major applications have been discussed below:

2.1. Educational Tools

QbE-STD is applied in educational settings to create interactive learning environments. For instance, students can use voice queries to retrieve relevant academic content, such as lecture recordings, tutorial videos, or supplemental materials, in real-time. MIT lecture browser [8] is an example of how QbE-STD is used for educational applications.

2.2. Audio Search Engines

QbE-STD enhances audio search engines by allowing users to search for specific spoken terms within large audio databases using natural language examples. This is very useful for researchers, students, and other people searching for specific information.

2.3. Voice-Activated Virtual Assistants

QbE-STD is integral to voice-activated virtual assistants, enabling users to interact by providing spoken examples of commands or queries. This is widely used in smart home systems and automobiles. These devices recognize spoken commands from users and execute corresponding actions or tasks in real-time.

2.4. Surveillance Systems

QbE-STD can be integrated into surveillance systems to automatically monitor live audio streams for specific keywords or phrases of interest. This can help detect security threats, identify safety concerns, or alert authorities to potential security threats in real time.

2.5. Content Monitoring and Compliance

Different organizations can check the online audio data for violations of certain policies. By automatically checking for predefined keywords or phrases, they can detect content violating the regulations.

2.6. Search in E-commerce Applications

QbE-STD can enhance search functionalities in e-commerce platforms, allowing users to find products or information by providing natural language examples of the items they are looking for.

2.7. Command and Control Systems

In military or industrial settings, QbE-STD can be used for command and control applications, allowing operators to provide spoken examples of commands or instructions for controlling systems.

2.8. Healthcare Applications

In healthcare settings, QbE-STD can assist medical professionals in accessing patient information, medical records, or clinical guidelines through voice-based queries. This facilitates hands-free interaction with electronic health records systems, improving efficiency and workflow in clinical environments.

2.9. Multimedia Retrieval

QbE-STD is used in multimedia retrieval systems to enable users to search for specific spoken content within large audio or video databases. This is valuable in various domains, including journalism, entertainment, and market research, for quickly locating relevant media assets based on spoken queries.

2.10. Call Center Analytics

In an environment like a call center, QbE-STD can be used to check customer and employee interactions. It will enable managers to check employee performance and identify trends.

There are numerous such real-time applications of QbE-STD. The effectiveness of QbE-STD makes it applicable in diverse contexts where rapid and accurate spoken content retrieval is essential.

3. Methodology

This study is based on a systematic review methodology. The method first examines and then assesses the existing research on QbE-STD. The research has been carried out in three stages: i) review preparation, ii) review execution, and iii) writing the summary of the review. Feature representation is a crucial step in QbE-STD, as it determines how speech signals are processed and compared. The goal of feature representation is to extract relevant information from raw audio waveforms while preserving phonetic and acoustic properties that distinguish spoken terms. Effective feature extraction improves system robustness against variations in speaker identity, background noise, and speaking styles.

4. Feature Representation Techniques

Feature representation is a crucial step in QbE-STD, as it determines how speech signals are processed and compared. The goal of feature representation is to extract relevant information from raw audio waveforms while preserving phonetic and acoustic properties that distinguish spoken terms. Effective feature extraction improves system robustness.

Traditional feature extraction methods rely on spectral analysis techniques, such as MFCCs and Perceptual Linear Prediction (PLP), which capture key acoustic characteristics of speech. These features have been widely used due to their simplicity and efficiency. More advanced methods leverage phonetic and subword-based representations, such as phone posteriorgrams and articulatory features, which provide a higher-level abstraction of speech content. Recent advancements in deep learning have led to the development of data-driven feature representations derived from Deep Neural Networks (DNNs).

4.1. Basic Feature Extraction Methods

Spectral features are basic speech features and are extracted from the speech utterances.

MFCC is one of the most widely used spectral features in speech processing. Speech utterance is first divided into smaller segments or frames (typically of size 20-25 ms) with an overlap of 10 ms to obtain the MFCC coefficients. First, compute the 13 MFCC coefficients for each frame, then obtain the delta and double delta coefficients. These delta coefficients provide information corresponding to the speech rate and acceleration of speech. Concatenate the 13 MFCC coefficients with delta and double delta coefficients to obtain a 39-dimensional feature vector for each frame [1].

In Linear Prediction (LP) analysis, an all-pole technique approximates the vocal tract spectral envelope. The weighted sum of the past samples can be used to predict the speech sample, and the weighted coefficients are called LP coefficients [1]. LPCC is a valuable parameter set that can be obtained from LP. LPCCs capture the vocal tract characteristics of the speaker. In PLP, the power spectrum is modified before applying the linear prediction all-pole model. PLP uses cubic compression of the spectrum. The all-pole model is also used by FDLP to describe the frequency components' temporal dynamics. It is observed that under noisy conditions, FDLP performs better than PLP [5]. To obtain the frequency domain representation in FBCC, the Fourier-Bessel transform is used instead of the Fourier transform. A damped sinusoid serves as the basis function for the FBCC and is more suited for spoken speech signals [10].

Similarity matches between the fundamental acoustic features may suffer from environmental noise mismatches; hence, rather than directly using these basic acoustic features, feature representations like phonetic or subword-based representations are used [11], which are explored in the next section.

4.2. Phonetic and Subword-Based Representations

Phonetic and subword-based representations aim to capture higher-level linguistic information from speech, moving beyond purely acoustic features like MFCCs or spectrograms. These representations help improve QbE-STD by enhancing robustness to speaker variability and background noise. The phonetic and subword-based representations are discussed in this section.

4.2.1. Phonetic Posteriorgrams

Phone Posteriorgrams (PPGs) are widely used phonetic representations that provide soft phonetic alignments for speech. They represent the probability distribution over phonetic units at each time frame. A phonetic posteriorgram involves plotting the posterior probabilities of phonetic classes versus time [12]. A neural network can be used to generate a phonetic posteriorgram. A neural network is trained to produce phone posteriorgrams using basic features, such as MFCC, extracted from the speech signal. [13]. A phonetic classifier is required to obtain a phonetic posteriorgram [14]. Building a phonetic classifier requires labeled data, so obtaining a posteriorgram is supervised and language-dependent [15]. It may be challenging to build a phonetic classifier for low-resource languages. One solution to this issue is to build phonetic classifiers using high-resource languages and use these to obtain phonetic posteriorgrams for low-resource languages [14]. However, phonetic posteriorgrams perform poorly when the test language is different from the language on which they are trained, as they cannot effectively capture the acoustic features of the target language [16].

4.2.2. Gaussian Posteriorgrams

Gaussian Posteriorgram representation is obtained using the Gaussian Mixture Model (GMM) [17]. MFCCs or FDLPs are grouped using GMM-based soft clustering. The number of components in the GMM should roughly equal the number of phonemes in the underlying language. The output of such a GMM is the posterior probabilities of its components corresponding to each speech frame and is called a Gaussian Posteriorgram (GP). Since this approach does not require labeled data, obtaining a posteriorgram representation is an unsupervised learning-based approach. GPs were first used in [9, 17]. GP is found to be an effective feature representation that suppresses speaker characteristics and allows multilingual search [5, 17, 18]. Instead of MFCC, FBCC has also been used to obtain GPs [10]. Instead of using a single GMM, the posteriorgram obtained from the mixture of GMMs is found to be a better representation as it captures the broad phonetic structure [19].

4.2.3. GAN-Based Posteriorgrams

The posteriorgrams obtained from the deep neural network, or GMM, use maximum-likelihood-based estimation and may affect the optimization of the network. Hence, a Generative Adversarial Network (GAN) generates the posteriors. GAN-based posteriorgrams represent an advanced phonetic feature extraction approach that improves upon traditional posteriorgrams by leveraging adversarial training. They have strong potential in low-resource, noisy, and cross-lingual spoken term detection tasks, making them valuable for QbE-STD applications. Initially, a GMM is trained with 39-d MFCC, and the GP is used as a target label for the GAN system. The input side is fed with contextual features, while the output side is provided with a labeled central frame posteriorgram. This considerably improves performance over posteriorgrams obtained from DNN [20].

4.2.4. Bottleneck Features

The bottleneck in a neural network is a layer with fewer neurons than the layers below or above. So, this layer consists of the compressed feature representations with the best fit in the available space. Bottleneck features represent a low-dimensional representation of data obtained from a neural network's bottleneck layer, which consists of the smallest number of hidden units [21]. Bottleneck features perform well for QbE-STD tasks [22-25]. Multilingual bottleneck features are used to implement multilingual STD [15, 26]. These are language-independent bottleneck features [21]. Specific approaches [27, 28] use the bottleneck features obtained from a high-resource language (cross-lingual bottleneck features) for search in a low-resource language. The bottleneck features are found to work well for QbE-STD.

CNN-based bottleneck features perform well even in noisy environments [29].

4.2.5. Articulatory Features

Phone classes are not the same for all languages, and hence, a more general representation, such as articulatory class information, is required. Articulatory classes allow cross-lingual speech recognition as they represent language-independent representations of speech sounds [30, 31]. They can be trained using relatively less data. The categorization of speech sounds into vowels and consonants, along with the articulators that are employed to characterize them, is known as the articulatory classes. It is found that Low Dimensionality Articulatory Motivated (LDAM) posteriorgrams have a representation that is unique for phonemes of various languages. However, when they are trained on a single language, they may only correctly represent some phonemes [32]. Hence, different approaches to joint training of multiple languages are explored in [30].

4.3. Deep-Learning-Based Representations

Deep learning has revolutionized speech processing by enabling the extraction of highly discriminative and context-aware representations from raw audio signals, unlike traditional handcrafted features such as MFCCs or PPGs, deep learning-based representations leverage neural networks to learn data-driven features that capture both acoustic and linguistic properties. This section discusses Acoustic Word Embeddings (AWEs) and transformer-based representations.

4.3.1. Acoustic Word Embeddings

AWEs are fixed-dimensional vector representations of spoken words, learned directly from speech signals. Unlike phonetic representations, which rely on explicit phoneme-level transcription, AWEs encode word-level acoustic similarities and can be trained in an unsupervised or supervised way. AWEs translate the speech segments into a fixed-dimensional vector space. In this representation, the distance between identical speech vectors is less, and between non-identical speech vectors is more. Preceding and successive words, when used as temporal context along with AWE, have been found to improve the performance of QbE-STD [33], and it also reduces runtime computation as dynamic-programming-based approaches like dynamic time warping (DTW) are not required. However, it requires a sufficient number of speech segment pairs. Deep convolutional neural network-based AWE is used for code-switching QbE-STD [33]. Instead of data from one language, it uses data from multiple languages for training.

Variable-length speech segments are mapped into fixed-length vectors using a Siamese Recurrent Autoencoder (RAE). The audio utterances are segmented into variable-length audio segments based on word boundaries. These segments are then fed into the Siamese RAE to obtain fixed-length vectors. The Siamese RAE receives word pairs with varied or identical word content in different instances. The Siamese RAE encoder’s last hidden state vector output, a feature vector for QbE-STD with related semantic content, is used as the output. Since the feature vectors are fixed-dimensional, matching becomes easier. This approach works well and reduces the detection time [34]. It adds the context frames of the desired spoken words to word pairings to produce fixed-length speech segment pairs. Multilingual bottleneck features are used to represent the word pairs. The speech segment pairings are then used to train a deep Bidirectional Long-Short-Term Memory (BLSTM) network with a triplet loss. The BLSTM backwards and forward outputs are concatenated to produce recurrent neural AWEs. During the searching step, the speech utterance and the query are converted into recurrent neural AWEs [27].

4.3.2. Transformer-based Representations

Transformers have emerged as a powerful deep-learning architecture for speech representation learning, offering self-attention mechanisms that capture long-range dependencies in audio. Transformer-based QbE-STD systems have shown significant advancements in recent research. [35-37] These systems use encoder-encoder structures with BERT-like encoders and modifications like convolutional layers, attention masking, and shared parameters to project recognized hypotheses and searched terms into a shared embedding space for scoring using calibrated dot products. Additionally, incorporating End-To-End (E2E) ASR systems can enhance performance by reducing Out-Of-Vocabulary (OOV) issues and improving search accuracy. Furthermore, deep convolutional neural network-based acoustic word embedding systems have been proposed for code-switching STD, combining audio data from multiple languages and applying variability-invariant loss for improved performance [33]. Attention-based pooling networks [38] are used for end-to-end QbE-STD systems. Audio utterances are initially segmented on word boundaries, and then an encoder with shared Recurrent Neural Networks (RNNs) is used to project audio utterances into hidden state sequences. This can be done offline. During the search process, similarity between the query and the reference utterance is determined using cosine distance, after extracting suitable features from both. End-to-end systems are also developed using attention-based multihop networks [37].

Speech from different people has different acoustic properties, and hence, in addition to using feature representations, additional techniques have been used to improve the effectiveness of QbE-STD. Speaker normalization is a technique used to eliminate speaker-specific details. Voice Tract Length Normalization (VTLN) is a widely used technique that nullifies variations resulting from the vocal tract length. VTLN [39] performs speaker normalization, and then these normalized features are used to obtain Gaussian posteriorgrams. VTLN normalized features improve the performance of the QbE-STD task. Posterior features can also be obtained from Deep Boltzmann Machines (DBMs). Semi-supervised and unsupervised techniques can be used to train a DBM. In the unsupervised technique, DBM is trained using labels generated from GMM. In contrast, in the semi-supervised approach, the unlabeled data is used to train DBM initially, and a small amount of labeled data is used to fine-tune it [40].

Feature representations using self-organizing maps (SOM) are used in [41]. The Affinity-kernel propagation approach is used to find the matching between the query and the reference utterance feature representations. [42] uses Wav2vec2.0 to learn representations, which are subsequently encoded as token sequences. For each token in this order, the Term Frequency-Inverse Document Frequency (TF-IDF) score is then calculated. Cosine similarity is used to compare the TF-IDF vector of the query and the TF-IDF matrix of the reference utterances.

In recent years, deep learning has emerged as a powerful approach for obtaining feature representations for QbE-STD. Deep learning techniques, such as CNNs, DNNs, and RNNs, are used to obtain more robust features from speech. Deep learning techniques enable the extraction of discriminative and contextually rich features directly from raw audio signals.

One of the key advantages of deep learning-based feature representations is their ability to automatically learn hierarchical and abstract representations of data, capturing intricate patterns and structures that may be difficult to extract using handcrafted features. These features are then used to measure the similarity between the query and audio utterances.

Additionally, architectures such as Siamese networks or triplet networks are utilized to learn similarity-preserving embeddings directly from pairs or triplets of audio segments, facilitating efficient comparison and retrieval of spoken terms.

Table 1 shows the comparative analysis of various feature representations used for QbE-STD.

Table 1. Comparative study of various feature representations used for QbE-STD

Feature representations	Advantages	Disadvantages	Used in
Phonetic Posteriorgram	Used to capture fine-grained phonetic information in speech. It represents posterior probabilities of phonetic units (e.g., phones)	Requires labeled data and alignment with phonetic transcriptions, and may not perform well when the target language is different from the trained language	[13, 14]
Gaussian Posteriorgram	Suitable for capturing fine-grained acoustic information, represents posterior probabilities of acoustic units (e.g., phonemes), does not require labeled data, and captures detailed acoustic information for sequence modeling tasks	Complexity in the feature extraction process requires alignment with text transcriptions.	[17, 43, 45]
GAN-based Posteriorgram	Used for generating synthetic speech representations, provides flexible and customizable feature representations that can capture diverse and realistic speech variations.	Complexity in training GAN models and the quality and fidelity of generated features may vary.	[20]
Bottleneck Features	Extracting from intermediate layers of neural networks, capturing discriminative information learned by neural networks, is effective for speech recognition tasks and works well even in noisy environments.	Complexity in training neural networks for feature extraction may require careful tuning of network architecture and training parameters.	[15,21,22, 25, 29]
Articulatory Features	Represent movements of speech articulators, allow cross-lingual speech recognition, can be trained using relatively less data, and capture detailed articulatory information.	Need to be trained on multiple languages to identify different phonemes correctly, limited availability of articulatory data, and complexity in data collection and feature extraction	[30,31]

Acoustic Word Embeddings	Embed words into a continuous vector space based on acoustic representations, and capture the semantic and contextual information of words in speech.	Complexity in training embedding models requires sufficient labeled data	[33, 46]
Transformer-based representations	Utilize self-attention mechanisms for capturing long-range dependencies, capturing global context and dependencies in speech.	Complexity in model architecture and parameter tuning, and large memory and computational requirements	[35-37]

Table 2. Similarity metrics used for computing a matching matrix between the query and reference utterance

Similarity Metric	Description/Advantages/Disadvantages	Used in
Cosine Similarity	• Computed as the cosine of the angle between the two vectors. • Efficient to compute, but less sensitive to subtle distribution differences	[5, 34, 47]
Kullback-Leibler Divergence (KLD)	• Similarity measure between probability distributions • Captures differences but is not symmetric and sensitive to noise	[45, 48]
Symmetric KLD	• Symmetric version of KLD • Removes directional bias but is computationally expensive	[19, 30, 39]
Histogram Intersection Kernel	• Computed as the sum of the minimum value between the features • Robust to noise but assumes normalized input	[43]
Log Similarity	• Log of the dot product • Highlights strong matches but needs normalization	[13, 15]

5. Matching Matrix Computation

A matrix is computed between the feature representations of the query and each reference utterance in the audio repository. Each value in the matrix represents a matching score. The query and the reference utterance are more comparable when the matching score is higher. A matching metric is employed to calculate these matching scores. The performance of QbE-STD depends on how effectively the metric captures the similarities between the feature vectors of the audio utterances. Various similarity metrics, such as cosine similarity and the symmetric Kullback-Leibler divergence (SKLD), have been explored in the literature.

Table 2 summarizes the various metrics used to compute the matching matrix. Cosine similarity is a simple, scale-invariant measure and gives good results for QbE-STD [5]. KLD and SKLD are suitable when GPs are used for QbE-STD, and KLD is not symmetric. They are sensitive to outliers and are computationally expensive measures. Kernel-based measures like HIK are computationally inexpensive and are suited for QbE-STD [43]. The choice of the similarity measure depends on the nature of the data and the characteristics of the feature representations. To choose an appropriate feature representation, it is necessary to consider factors such as robustness to noise, performance, and computational efficiency.

6. Similarity Detection Techniques

After the matching matrix is computed, a suitable similarity detection technique is employed to assess the relevance of the query to the reference utterance. This step is crucial in determining whether the spoken query appears in the reference audio. Various approaches have been explored in the literature to effectively perform this alignment and similarity scoring.

Most existing techniques use the DTW algorithm over the computed matching matrix, and the DTW score is used to determine the relevance of the query to the reference utterance. It calculates the DTW matrix based on the previously computed matching matrix and determines whether the query is relevant to the utterance using a predefined threshold. DTW [18] is used to find the optimal alignment between two sequences satisfying some conditions. Warping constraints based on starting and endpoints, locality, slope weightage, and monotonicity are applied in DTW [44]. The standard DTW technique may not be efficient for searching large datasets [27]. Various changes have been proposed to increase the speed of computations.

Variations can depend on local weights, global constraints, and step size. Step size conditions will restrict the slope of the warping paths. Local weights will favor some directional alignments, like diagonal, vertical, and horizontal. Various adjustments to the DTW algorithm have been suggested based on these variations.

Segmental DTW (S-DTW)[17] involves dividing the speech utterance into smaller fragments and performing DTW on each segment. S-DTW uses global constraints to restrict the alignment of the spoken audio segments, which may be computationally expensive. Non-Segmental DTW (NS-DTW) [5] uses local rather than global constraints. [49] proposed a fast and memory-efficient DTW (MES-DTW) algorithm that suggested modifications to the subsequence

DTW. It uses the query's fixed start and end points and then searches for matching subsequences. Fast NS-DTW (FNS-DTW) [5] uses reduced GPs for QbE-STD. Slope-constrained DTW [17] enforces restrictions on the slope of the warping path. Subspace-regularized DTW [15, 50] regularizes the matching matrix using the test utterance and the query's subspace structure. It uses sparsity and DTW to develop an effective system.

Convolutional Neural Network (CNN) is widely used in image classification. The success of CNN in image classification tasks motivated its use for QbE-STD. CNN is a deep-learning algorithm that is very useful in pattern discovery. CNN does not need handcrafted features but can learn features on its own. It consists of a cascade of convolution and pooling layers, followed by fully connected layers. The CNN is used to categorize images into two classes—positive and negative—after converting the matching matrix that was created in QbE-STD into an image. Whereas the negative class denotes that the query is absent from the utterance, the positive class shows that the query is present in the reference utterance [43]. The methods may use different feature representations and metrics to compute the matching matrix. All these methods convert the matching matrix to an image and use CNN for QbE-STD.

In [15], bottleneck features are used, and modified cosine similarity is used to construct the matching matrix. This matching matrix is then converted to an image. If the query appears in the utterance, the diagonal entries in the similarity matrix will be similar, resulting in a quasi-diagonal pattern. So, the presence of the query in the test utterance is depicted by a quasi-diagonal pattern representing a positive class. The absence of a diagonal pattern represents the negative class. CNN takes the entire image (or the matching matrix) together to locate the pattern, compared to the DTW algorithm, which makes local decisions. Hence, an end-to-end system [21] using CNN is better than one using DTW.

While [15] uses modified cosine similarity to compute the similarity matrix, [43] uses kernel-based matching for computing the similarity matrix. HIK is used to compute the similarity score and is a faster method. DTW with CNN is used for QbE-STD in [45], and this approach uses a modified DTW and visualizes the warping matrix as a grayscale image. The matching score between features was calculated using KLD, and a CNN was then trained on these images to classify keywords using the texture of the warping matrix image. In [51], the matching matrix is computed using symmetric KL divergence. Then, image processing methods like area filtering, edge detection, edge filling, and line dilation are applied to the similarity matrix to identify the probably matched regions, followed by the angle histogram technique to obtain the matched regions. Finally, the frame histogram technique is applied to divide the images further into positive and negative class images, which are used to train a CNN. The matched region images obtained from the image processing methods are given to the CNN. The study referenced in [36] utilized transformer architecture to analyze Spoken Term Detection (STD). The encoder component of the transformer extracts context-dependent vector representations of the input. Both the reference utterance and the input query are transformed into a shared embedding space. The sigmoid-calibrated dot product is used to compute the similarity between these two vectors. In contrast, the approach detailed in [52] employs a Fully Connected Convolutional Neural Network (FC-CNN) to determine whether the speech stream contains the query. In this method, the reference utterance is input into a CNN with an attention mechanism after the query is appended to it.

DTW is the state-of-the-art technique used for matching. Different variants of DTW are used to increase the effectiveness of QbE-STD. DTW effectively captures patterns and temporal variations within sequences, but it is computationally expensive and can be sensitive to outliers. Recent techniques have focused on the use of deep learning techniques. The techniques visualize the matching matrix as an image and then use CNN as a classifier. The metric and technique used to obtain the matching matrix are different. Some approaches use image processing techniques before classification to lower the number of false alarms and missed detections. Other approaches use RNNs, RAEs, and transformer-based architectures to convert the features of the query and the utterance to a fixed dimension. The fixed-dimensional feature vectors of the query and the utterance are matched using a suitable matching metric. QbE-STD using CNN performs well and depends on the similarity measure used to calculate the matching matrix. The choice of a matching technique depends on the nature of the data and the availability of computational resources. If the dataset is large, then CNN may be a good choice for automatic feature learning. Next, the various datasets used for QbE-STD are discussed.

7. Datasets used for QbE-STD

Different datasets are available and used for QbE-STD. Some datasets are developed explicitly for QbE-STD due to benchmarking initiatives, while others are for various speech-related activities. In this section, a brief discussion of every dataset is presented.

7.1. TIMIT Dataset

The TIMIT acoustic-phonetic continuous speech corpus [53] provides spoken data for developing and evaluating ASR systems. It consists of recordings of 630 speakers. These speakers speak eight distinct American English dialects. Each speaker reads ten sentences that are rich in phonetics. It is a phonetically balanced dataset that provides word-level and phoneme-level speech transcriptions. TIMIT is not explicitly designed for QbE-STD. Queries can be extracted from the given sentences using the metadata provided along with the sentences. The data is then divided into train and test datasets. This dataset is used for QbE-STD in [17, 39, 44, 45].

7.2. MediaEval 2011 SWS Dataset

MediaEval 2011 (SWS) [54] is a dataset developed by the Spoken Web center at IBM Research, India, for the MediaEval 2011 Spoken Web Search (SWS) task. The audio content is a spontaneous speech from low-literate users collected from mobile phone communications. It contains audio recordings from four Indian languages: Gujarati, English, Telugu, and Hindi [55]. MediaEval datasets are a part of the MediaEval benchmarking initiative for spoken web search tasks.

7.3. MediaEval 2012 SWS Dataset

MediaEval 2012 (SWS) [56] was made available as a part of the MediaEval benchmarking initiative for multimedia evaluation 2012. It contains two datasets: one is an Indian dataset from four different Indian languages, namely English, Gujarati, Hindi, and Telugu. The other is the African dataset, which consists of audio content in 11 South African languages [5, 10, 57].

7.4. MediaEval 2013 SWS Dataset

MediaEval 2013 (SWS) dataset [58] consists of audio files from different languages and acoustic conditions. It contains data from nine languages. These nine languages cover the European and African language families. It consists of 20 hours of audio recordings divided into development and evaluation sets of queries [13, 39, 47, 48, 59]. This dataset was developed explicitly for STD tasks.

7.5. MediaEval 2014 QUESST Dataset

MediaEval 2014 Query by EXAMPLE Search on Speech (QUESST) database [60] consists of audio files from various languages under different acoustic conditions. It consists of 23 hours of audio recordings. It considers approximate matching along with exact matching. There are three types of matching: Type 1 (exact), Type 2 (variant), and Type 3 (reordering or filler). Type 1 means an exact match; type 2 means it is not an exact match, but there could be slight variations either at the beginning or end of a query; and type 3 again is not an exact match, but it requires all the words in a query but the order of words may be different **[**13, 15, 48].

7.6. MediaEval 2015 QUESST Dataset

MediaEval 2015 (QUESST) dataset [61] consists of audio files from a large set of languages. The speech corpus contains audio with heavy accents recorded in challenging acoustic conditions. This dataset comprises around 18 hours of speech (11662 files) in the following 7 languages: Slovak, Albanian, English, Czech, Mandarin, Portuguese, and Romanian. The QUESST 2015 dataset consists of 447 evaluation and 445 development queries. The number of queries per language is uniform. Like the MediaEval 2014 database, it consists of three types of queries [62, 63]. This dataset was developed explicitly for STD tasks.

7.7. Globalphone CORPUS

Karlsruhe Institute of Technology (KIT) has built a database called GlobalPhone corpus [64, 65]. It contains multilingual data, i.e., data from 20 different languages. The critical property of this corpus is that it is designed to be balanced in terms of audio or text data per language, audio data quality, the collection scenario, and transcription conventions [21, 50]. This dataset is not explicitly designed for QbE-STD**.**

7.8. AMI Meeting Corpus

The Augmented Multi-Party Interaction (AMI) meeting corpus [66, 67] consists of approximately 100 hours of meeting recordings and supports multidisciplinary research. Both real-time and scenario-driven meeting recordings are present in this corpus. English language recordings of the meetings are made in various acoustic settings [50]. The AMI meeting corpus is designed explicitly for the AMI project and for developing meeting browsers, but is used for various research purposes. It is used for the QbE-STD task in [50].

7.9. LWAZI Corpus

LWAZI speech corpus [68, 69] contains telephone speech. The data consists of a speech from eleven official languages of South Africa and includes approximately 200 speakers per language [31]. The languages available are Xitsonga, Afrikaans, isiNdebele, English, isiZulu, isiXhosa, siSwati, Sepedi, Tshivenda, Sesotho, and Setswana. This dataset was not explicitly developed for QbE-STD but has been used in [31] for the QbE-STD task..

7.10. SHRUTI Corpus

SHRUTI is also a read-speech corpus in Bengali designed for ASR systems. It consists of approximately 7383 unique sentences spoken by 34 native speakers. The sentences cover most of the frequently spoken words in the Bengali language [30]. This corpus was designed to develop and evaluate ASR systems but is also used for QbE-STD tasks [30].

7.11. MIT Lecture Corpus

The MIT Lecture corpus [8, 70] contains approximately 300 hours of audio data. This data is recorded from lectures on eight subjects and various seminars. Most data is recorded in a classroom and may contain non-speech artifacts. This dataset was not explicitly designed for QbE-STD but was used in [17] for QbE-STD.

7.12. English Switchboard Corpus

The English Switchboard Corpus contains approximately 260 hours of speech [71]. The data consists of approximately 2400 telephone conversations among 543 speakers from various regions of the United States. This corpus is available for free download. A computer-driven robot operator system was used to collect the data. This dataset was designed for ASR systems but is used for the QbE-STD task in [26, 27, 46].

Our study explored different datasets for QbE-STD. These datasets include Medieval datasets that are designed explicitly for QbE-STD. Mediaeval datasets provide separate sets of development, evaluation, utterances, keywords and scripts to perform performance evaluations. It is always advisable to use these datasets since they are simple and are part of the Medieval benchmarking system.

8. Metrics Used for Evaluation

Various metrics are employed to assess the QbE-STD system. Some QbE-STD systems output the time location of the query within the speech utterance, while the other systems identify whether the query is present in the utterance. Some detection is associated with a score and detection threshold Ɵ.

If the score is greater than the detection threshold Ɵ, it is considered a hit; otherwise, it is a miss. Let TP stand for the number of true positives, or examples correctly identified as belonging to a positive class, and TN for the number of true negatives, or occurrences correctly classed as belonging to a negative class. Let us say that False Positive (FP) represents the number of false positives, or cases mistakenly classified as positive, and False Negative (FN) represents the number of instances mistakenly classified as negative. The various evaluation metrics are discussed in detail below.

8.1. Accuracy and Error Rate

The ratio of properly categorized examples to all occurrences is known as accuracy [1]. Ideally, the accuracy should be 100%, and the error rate must be 0%.

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{1}$$

$$ErrorRate = \frac{FP+FN}{(TP+TN+FP+FN)} \tag{2}$$

Equations (1) and (2) show the formula for accuracy and Error rate using the true and false positives and negatives.

8.2. Miss Rate and False Alarm Rate

The miss rate is the ratio of the number of missed positive instances to the total number of positive instances. It measures missed detections [1]. It is defined as

$$MissRate = \frac{FN}{(TP+FN)} \tag{3}$$

The false alarm rate is the number of missed negatives among all negative instances and is defined as

$$FAR = \frac{FP}{(TN+FP)} \tag{4}$$

Equations (3) and (4) show the formula for miss rate and False Alarm Rate (FAR).

8.3. Recall and Precision

The precision is the ratio of correct positive predictions, i.e., true positives, to all the items predicted to be positive. It gives a confidence measure concerning positive predictions of the system.

All predictions are taken into account by precision, but occasionally, only the top N results of the system are considered; this is known as P at N (P@N) [1, 14, 17, 46]. The ratio of correctly predicted positive classes to all of the actual positive items is known as the recall.

$$Pr\,e\,cision = \frac{TP}{(TP+FP)} \tag{5}$$

$$Re\,c\,all = \frac{TP}{(FN+TP)} \tag{6}$$

Equations (5) and (6) show the formulas for precision and recall, respectively.

8.4. Mean Average Precision (MAP)

The ratio of accurately predicted positive classes is known as precision. The Mean Average Precision (MAP) is the average precision for each query. The mean of the precision values for the top k papers is used to get the average precision [1, 46].

8.5. DET Curve and Equal Error Rate (EER)

The performance of detection tasks involving a tradeoff between miss and false alarm rates is represented by the Detection Error Tradeoff (DET) curve [72]. For every query, the average missed detection and false alarm rates are determined. A DET curve is then created by plotting these errors against one another for various threshold settings [13]. The point on the DET curve where the false acceptance rate and the false rejection rate are identical is known as the Equal Error Rate (EER) [17].

8.6. Term Weighted Value (TWV)

Let q be the query. Let $N_{act}(q)$ be the actual occurrences of the query q in the speech utterance. Let $N_H(q,\theta)$ be the number of hits of the query for a threshold θ. Let $N_{NO}(q)$ be the number of non-occurrences of a query q and is defined as $N_{NO}(q)=n_{tps}$ X T_{audio} -$N_{act}(q)$, where T_{audio} is the total duration of the speech utterances in seconds and n_{tps} is the number of trials per second. This metric is typically used for QbE-STD systems, which output the time location of the query. Typically, n_{tps} =1. Let $N_F(q,\theta)$ be the number of false alarms of a query q for a thresholdθ. The weighted combination of the miss rate and the false alarm is then averaged over a set of all queries, and is the Term Weighted Value (TWV) [12].

$$TWV(\theta) = 1 - \frac{1}{(|Q|)} \sum_{\forall q \in Q} [PM(q,\theta) + \beta PFA(q,\theta)] \quad (7)$$

Where $P_M(q, \theta)$ is the miss probability of a query q for a given detection threshold $\theta$and is obtained as

$$PM(q,\theta) = 1 - \frac{NH(q,\theta)}{Nact(q)} \quad (8)$$

And $P_{FA}(q,\theta)$ is the false alarm probability of a query q and is given by

$$PFA(q,\theta) = \frac{NF(q,\theta)}{NNO(q)} \quad (9)$$

and β is the weight factor, greater than zero and is defined as

$$\beta = \frac{CFA(1 - Pt\,arg\,et)}{CM(q) \times Pt\,arg\,et} \quad (10)$$

Where $C_{FA} > 0$ and $C_M > 0$ are the costs of false alarms and miss errors, respectively [12]. P_{target} is the prior probability of a query q and can be computed as

$$PT\,arg\,e\,t = \frac{Nact(q)}{Taudio} \quad (11)$$

The largest value that can be obtained for TWV is 1, representing a perfect system output[1].

The average term weighted value TWV obtained by a QbE-STD system for a given threshold. The maximum term weighted value is the maximized TWV and does not depend on the threshold $\theta$of the search system. MTWV is thus the preferred metric [13,73]. The TWV on the DET curve, where a value yields the maximum TWV, is known as the MTWV. The NIST has defined three evaluation measures for STD assessment: TWV, ATWV, and MTWV [74].

8.7. Normalized Cross Entropy

Normalized cross entropy C_{nxe} is another metric used for the evaluation of the QbE-STD system. Cross-entropy represents the expected value of information. C_{nxe}=0 for an ideal system. C_{nxe}=1 for a system with no informative value, whereas $C_{nxe} > 1$ would indicate an error in the log-likelihood ratio scores [13].

9. Benchmarking Platforms

Many techniques for searching audio have been developed by researchers utilizing different evaluation measures and databases. Comparing the performance of these systems is complex, and hence, many benchmarking systems are created. These benchmarking platforms include assessment metrics in addition to development and test datasets [1]. Some of the benchmarking platforms are discussed in this section.

9.1. NIST

Open KWS was launched by the National Institute of Standards and Technology (NIST) to promote research in STD. The goal of this program is to create high-performing KWS systems on a new language quickly [1, 72].

The OpenKWS project is a follow-up to the 2006 STD evaluation, which uses broadcast news recordings in English, Mandarin, and Arabic, conversational telephone speech (CTS), and conference meeting data to test KWS algorithms. Every year, participants will receive resources for testing, training, and development in CTS; nevertheless, they will only have a short time window to construct their systems. The results of the evaluation are then discussed at the evaluation workshop [75].

9.2. MediaEval

MediaEval aims to assess novel algorithms for retrieving and accessing multimedia. Participants who are interested in multimodal approaches to multimedia are drawn to MediaEval. MediaEval, a community-driven standard, is managed by the MediaEval organizing committee. It is made up of the organizers of each assignment for that year.

9.3. Spoken Web Search (SWS)

In 2011, 2012, and 2013, MediaEval conducted Spoken Web Search (SWS) tasks [54]. The SWS task is to search for an audio query in another audio file, making it challenging for the researchers. The task is to build a language-independent system to find the audio file containing the query and the query location in the audio file. The evaluation process was conducted per the STD evaluation guidelines provided by NIST.

9.4. Query by Example Search on Speech Task (QUESST)

In 2014, “Query By Example Search On Speech Task (QUESST)” replaced the term “Spoken Web Search” (SWS) [60]. The suggested queries in QUESST were what made it novel. QUESST challenge, in addition to standard single and multiword searches, also includes complex single and multiword queries. Three different kinds of queries were defined, depending on the complexity. Type 1 (T1) comprised questions that could start or end slightly differently. “Researcher” matching “research” in the audio utterance is an example of a type 1 query [60]. Type 2 (T2) questions are those in which the order in which words in the query appear in the search utterance varies. Similar to type 2 searches, Type 3 (T3) queries permit filler text to be placed between the multiple matched terms in the reference utterance. An example of a T3 query is that the query “black cat” must match the utterances “My cat is black” and “I have a black and cute cat.”

9.5. Albayzin Evaluation

The Special Interest Group on Iberian Languages of the International Speech Communication Association (ISCA) and the Spanish Network of Speech Technologies jointly support the Albayzin evaluation [76]. It provides a mechanism to promote research on speech tasks [77]. There are two tasks defined in this evaluation: STD and QbE-STD. Three different speech databases on different domains were used in the Albayzin 2020 evaluation [78]. The first database, the MAVIR database, comprises the talks from the workshops; the second database, the RTVE database, comprises the news broadcast programs; and the third database, the SPARL20 database, includes the Spanish parliament sessions. In-depth post-evaluation assessments based on particular query properties (in- and out-of-vocabulary, single- and multiword, native and foreign) are also included in the Albayzin 2020 evaluation. The data augmentation method for the STD task and an end-to-end system for the QbE STD were the innovative features of the submitted systems.

A detailed list of these benchmarking platforms is summarized in Table 3.

Table 3. List of audio search benchmarking platforms

Benchmarking platform	Evaluation year
NIST STD	2006
NIST (Openkws)	2013, 2014, 2015, 2016
MediaEval (SWS)	2010, 2011, 2012, 2013
MediaEval (QUESST)	2014, 2015, 2016
Albayzin Evaluation	2006, 2008, 2010, 2012, 2014, 2016, 2018, 2020, 2022, 2024

10. Challenges and Future Directions

QbE-STD cover various aspects of the technology, including scalability, accuracy, robustness, and usability. As with any evolving field, QbE-STD also faces persistent challenges. Some challenges faced are variability in pronunciation, presence of noise in the utterances, providing real-time response, and developing QbE-STD for low-resource languages. Audio data is highly variable, and hence, there is a need for a reliable feature representation that can effectively capture the spoken term irrespective of gender, dialect, and age variations. The audio database has utterances recorded in a good environment, but the query provided by the user may be in a noisy environment. The matching methods may not work well in the presence of noise. Hence, the feature representations and the matching methods need to handle the noise in the data. As audio datasets grow, QbE-STD systems must efficiently scale to handle large volumes of data without compromising performance. Efficient indexing and retrieval techniques may be used to address scalability. The non-availability of labeled data for low-resource languages may make developing an efficient QbE-STD system challenging. Transfer learning techniques may be explored where models pre-trained on large datasets for related tasks are fine-tuned on smaller, domain-specific datasets. Active learning strategies can be employed. Real-time QbE-STD systems are essential for applications such as live broadcasting, call center analytics, and voice-controlled devices. Future research may explore efficient algorithms and hardware accelerations to enable real-time processing of audio streams with low latency. Addressing these challenges requires a holistic approach, combining machine learning and signal processing advancements.

11. Discussion

In this work, QbE-STD is investigated as it offers a dynamic field with a wide range of approaches and changing research directions. Enabling effective retrieval of spoken terms from massive audio corpora is the core objective of QbE-STD, and our review identifies critical perspectives and issues in this field. One notable trend observed is the increasing use of machine learning techniques in QbE-STD. From DTW, the shift is towards using CNN, LSTM, and RAE in QbE-STD. Deep learning techniques allow automatic, robust feature learning, which extracts discriminative features from unprocessed input.

Feature representations play a crucial role in the effectiveness of QbE-STD systems. Different representations, ranging from Gaussian, phonetic posteriorgrams, BNF, AWEs, and transformer-based representations, are analyzed. The choice of the representation depends on the nature of the data and interpretability. The detailed discussion on these representations will allow the researchers to choose a suitable representation wisely.

Similarity metrics used to calculate matching matrices between query and utterance feature representations are discussed. Similarity measures explored are cosine similarity, Dynamic Time Warping (DTW), Symmetric Kullback-Leibler Divergence (SKLD), Kullback-Leibler Divergence (KLD), and Histogram Intersection Kernel (HIK). Each measure has advantages and disadvantages, highlighting the need to choose the appropriate metric based on the nature of the data and task requirements.

12. Conclusion

QbE-STD consists of feature representation followed by template matching. The datasets available for QbE-STD are summarized in Table \ref{tab1} for easy reference. Table \ref{summary} provides a comparative study of feature representations, matching techniques, datasets, and evaluation measures used for QbE-STD.

This work presents a systematic review of techniques used for QbE-STD. The major steps in QbE-STD are feature representation and similarity detection. After the features are extracted and represented, matching is performed between the query and the utterance. The paper first discusses the various feature representations, followed by matching techniques. Feature representations include basic spectral features, posteriorgrams, bottleneck features, AWE, and transformer-based representations. After representing the features, a matching matrix is computed between the query and the utterance using a suitable distance metric. Table \ref{tab2} summarizes the different distance metrics used for QbE-STD. After the matching matrix is computed, matching techniques determine whether the query is present in the utterance. Matching techniques include the state-of-the-art DTW algorithm and CNN. Numerous enhancements have been made to the DTW algorithm to enhance the performance of the QbE-STD system. Moreover, end-to-end systems, which convert features into fixed-length vectors, are explored alongside attention mechanisms. The paper also discusses in detail the various databases available for QbE-STD. It also covers the different evaluation measures and benchmarking platforms used for QbE-STD.

STD has been a research topic in the speech community for a long time, and now the emphasis has shifted mainly to low-resource languages with less data available. The techniques shift towards using deep neural networks for QbE-STD. Integrating more sophisticated deep learning architectures is likely to improve the learning of feature representations further, contributing to higher performance of QbE-STD systems. End-to-end systems need to be explored in detail for improved accuracy. Future research directions are required to improve the performance of QbE-STD for low-resource languages and in real-time environments. Multimodal approaches that require audio and visual information fusion may also be explored for video data. Combining QbE-STD with video context information may increase the system's performance. Advancements in zero-shot and few-shot learning techniques will enable QbE-STD systems to recognize spoken terms with minimal training examples. This approach is helpful for low-resource languages with less data available. Also, most of the techniques do not address the issues where the length of the utterance may be very large as compared to the length of the query. Table 4 presents the comparative analysis of different matching techniques used for QbE-STD and their performances.

In conclusion, this review consolidates a comprehensive understanding of QbE-STD, highlighting the dynamic interaction between feature representations, similarity measures, and machine learning techniques. As the field continues to evolve, it is critical to be aware of new developments and address the persistent challenges to advance QbE-STD toward better performance, effectiveness, and applicability in real-world scenarios. This paper serves as a roadmap for researchers and practitioners navigating this field of QbE-STD.

Table 4. Comparative study of matching techniques and datasets used in various QbE-STD papers

References	Matching Technique	Dataset	Performance
[17]	Log of the dot product and SDTW	1) TIMIT 2) MIT Lecture Corpus	1) P@N=50% and EER=15% 2)P@N=39.3% , EER=15.8%
[7]	Log of the dot product and DTW	NIST 2006	P@10=66.3%, P@N=54.7%, EER=9.8%
[16]	NSDTW	SWS 2013	MTWV (dev) =0.2765, MTWV (eval)=0.2413
[47]	Log of the Cosine dist. and DTW	SWS 2013	MTWV=0.464
[5]	Log of Cosine dist. And NSDTW	MediaEval 2012	MTWV=0.399 (dev), Miss Probability=0.426, FAR=0.01136
[31]	Log of cosine dist. And NSDTW	MediaEval 2012	MTWV=0.494 (dev), MTWV=0.492 (eval)
[73]	Log of Cosine dist. And NSDTW	MediaEval 2012	MTWV=0.489 (dev), MTWV=0.469 (eval)
[48]	Symmetric KLD and DTW	SWS 2013, QUESST 2014	MTWV= 0.386 (dev), MTWV=0.359 (eval)
[45]	KLD, DTW and CNN	TIMIT	FAR=0.0752, FRR=0.0758, Overall Error Rate=0.075

[13]	Log of the dot product and CNN	SWS 2013, QUESST 2014	MTWV(SWS)=0.388, Normalized Cross Entropy=0.6028, MTWV (QUESST)=0.5853
[50]	Cosine Similarity	SWS 2013, AMI Meeting Corpus	MTWV=0.3020 (10 ex. per query), and MTWV=0.4362 (1 ex. per query)
[43]	HIK and CNN	TIMIT	FAR = 0.038, miss rate = 0.042 (64-GMM)
[46]	Cosine distance and CNN	English Switchboard Corpus	MAP=0.502, P@5=0.567, P@N=0.462
[34]	Cosine distance	TIMIT, Real Scene Chinese Speech data	MAP=0.116,0.234
[19]	SKLD and DTW	TIMIT and SWS 2013	MTWV=0.494 (dev), MTWV=0.453 (eval)
[51]	SKLD and Hierarchical clustering	IITKGP-SDUC Bengali Speech db	Hindi: Accuracy=48.89, Bengali : accuracy=70.64
[39]	SKLD and DTW	TIMIT, SWS 2013	MTWV=0.456(dev), MTWV=0.412(eval)
[30]	SKLD and DTW	Shruti corpus	MAP=61.99, P@N=59.23
[20]	SKLD	TIMIT	MAP=29.42
[21]	Log of the dot product and CNN	SWS 2013, QUESST 2014	MTWV=0.6499
[36]	Sigmoid-calibrated dot product	USC-SFI	Monolingual setup: ATWV=0.7938(English), ATWV=0.9120(Czech), Multilingual setup: ATWV=0.7925(English), ATWV=0.9062(Czech)
[41]	Acoustic feature map and affinity kernel propagation	QUESST 14	: MTWV=0.3841 (dev), MTWV=0.3796 (eval)
[79]	Self-Organizing Maps	Zero-resource speech corpus	For English queries: Type Recall:17.1, Token recall=24.7, Boundary Recall=72.7
[80]	Cosine Similarity, CNN and LSTM	Librispeech and TIMIT	MAP=70.22%, P@5=80.62%
[37]	Attention-based multihop networks	Librispeech corpus	MAP=0.6789 (Test1), MAP=0.6430 (Test2), MAP=0.5830 (Test3)
[33]	Cosine similarity with Deep CNN	Librispeech corpus (English) and Chinese data	MAP=0.795, P@N=0.464, P@5=0.820
[42]	TF-IDF on Wav2Vec2.0 model	Hindi and Librispeech corpus (English)	English: MAP=0.55, ATWV=0.55, Hindi: MAP=0.69, ATWV=0.62
[81]	HIK and CNN	TIMIT, SWS 2013	TIMIT: Accuracy=98%, SWS 2013: Accuracy=75%
[82]	HIK and CNN	TIMIT	Accuracy=96.97%,

References

[1] Leena Mary, and G. Deekshitha, *Searching Speech Databases: Features, Techniques and Evaluation Measures*, *Springer International Publishing*, pp. 1-76, 2018. [Google Scholar] [Publisher Link]

[2] J.S. Bridle, "An Efficient Elastic-Template Method for Detecting Given Words in Running Speech," *British Acoustical Society Spring Meeting*, pp. 1-4, 1973. [Google Scholar]

[3] A. Higgins, and R. Wohlford, “Keyword Recognition Using Template Concatenation,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tampa, FL, USA, vol. 10, pp. 1233-1236, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[4] Roy Wallace, Robbie Vogt, and Sridha Sridharan, “A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation,” *Interspeech*, pp. 2385-2388, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[5] Gautam Mantena, Sivanand Achanta, and Kishore Prahallad, “Query-by-Example Spoken Term Detection using Frequency Domain Linear Prediction and Non-Segmental Dynamic Time Warping,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 5, pp. 946-955, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kishan Thambiratnam, and Sridha Sridharan, “Rapid Yet Accurate Speech Indexing Using Dynamic Match Lattice Spotting,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 346-357, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[7] Timothy J. Hazen, Wade Shen, and Christopher White, “Query-by-Example Spoken Term Detection using Phonetic Posteriorgram Templates,” *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, Moreno, Italy, pp. 421-426, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[8] James Glass et al., “Analysis and Processing of Lecture Audio Data: Preliminary Investigations,” *Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT-NAACL 2004*, Boston, Massachusetts, USA, pp. 9-12, 2004. [Google Scholar] [Publisher Link]

[9] S. Davis, and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357-366, 1980. [CrossRef] [Google Scholar] [Publisher Link]

[10] Drisya Vasudev et al., “Query-by-Example Spoken Term Detection using Bessel features,” *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Kozhikode, India, pp. 1-4, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[11] Haipeng Wang et al., “Using Parallel Tokenizers with DTW Matrix Combination for Low-Resource Spoken Term Detection,” *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, pp. 8545-8549, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[12] L.J. Rodriguez-Fuentes, and Mikel Penagarikano, “*MediaEval 2013 Spoken Web Search Task: System Performance Measures*,” Technical Report Department of Electricity and Electronics, University of the Basque Country, pp. 1-14, 2013. [Google Scholar] [Publisher Link]

[13] Dhananjay Ram, Lesly Miculicich, and Hervé Bourlard, “CNN-based Query by Example Spoken Term Detection,” *Interspeech*, pp. 92-96, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Vikram Gupta et al., “A Language-Independent Approach to Audio Search,” *Interspeech*, pp. 1125-1128, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[15] Dhananjay Ram, Lesly Miculicich, and Hervé Bourlard, “Multilingual Bottleneck Features for Query by Example Spoken Term Detection,” *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Singapore, pp. 621-628, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Gautam Mantena, and Kishore Prahallad, “IIIT-H SWS 2013: Gaussian Posteriorgrams of Bottle-Neck Features for Query-by-Example Spoken Term Detection,” *MediaEval 2013 Workshop*, Barcelona, Spain, pp. 1-2, 2013. [Google Scholar] [Publisher Link]

[17] Yaodong Zhang, and James R. Glass, “Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams,” *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, Moreno, Italy, pp. 398-403, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[18] M. Muller, *Dynamic Time Warping*, Information Retrieval for Music and Motion, pp. 69-84, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[19] Maulik C. Madhavi, and Hemant A. Patil, “Design of Mixture of GMMs for Query-by-Example Spoken Term Detection,” *Computer Speech & Language*, vol. 52, pp. 41-55, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Neil Shah et al., “Query-By-Example Spoken Term Detection Using Generative Adversarial Network,” *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Auckland, New Zealand, pp. 644-648, 2020. [Google Scholar] [Publisher Link]

[21] Dhananjay Ram, Lesly Miculicich, and Hervé Bourlard, “Neural Network Based End-to-End Query by Example Spoken Term Detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1416-1427, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Hongjie Chen et al., “Unsupervised Bottleneck Features for Low-Resource Query-by-Example Spoken Term Detection,” *Interspeech*, pp. 923-927, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[23] Emre Yilmaz, Julien van Hout, and Horacio Franco, “Noise-Robust Exemplar Matching for Rescoring Query-by-Example Search,” *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, pp. 1-7, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Julien van Hout et al., "Tackling Unseen Acoustic Conditions in Query-by-Example Search Using Time and Frequency Convolution for Multilingual Deep Bottleneck Features," *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, pp. 48-54, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[25] Miroslav Skácel, and Igor Szöke, "But Quesst 2015 System Description," *CEUR Workshop Proceedings*, pp. 1-3, 2015. [Google Scholar] [Publisher Link]

[26] Yougen Yuan et al., "Pairwise Learning Using Multi-Lingual Bottleneck Features for Low-Resource Query-by-Example Spoken Term Detection," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, pp. 5645-5649, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Yougen Yuan et al., "Query-by-Example Speech Search Using Recurrent Neural Acoustic Word Embeddings with Temporal Context," *IEEE Access*, vol. 7, pp. 67656-67665, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Yougen Yuan et al., "Fast Query-by-Example Speech Search Using Attention-Based Deep Binary Embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1988-2000, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[29] Hyungjun Lim et al., "CNN-based Bottleneck Feature for Noise Robust Query-by-Example Spoken Term Detection," *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala Lumpur, Malaysia, pp. 1278-1281, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[30] Abhimanyu Popli, and Arun Kumar, "Multilingual Query-by-Example Spoken Term Detection in Indian Languages," *International Journal of Speech Technology*, vol. 22, pp. 131-141, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] Gautam Mantena, and Kishore Prahallad, "Use of Articulatory Bottle-Neck Features for Query-by-Example Spoken Term Detection in Low Resource Scenarios," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, pp. 7128-7132, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[32] Abhimanyu Popli, and Arun Kumar, "Capturing Indian Phonemic Diversity with Multiple Posteriorgrams for Multilingual Query-by-Example Spoken Term Detection," *2017 Twenty-Third National Conference on Communications (NCC)*, Chennai, India, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[33] Murong Ma et al., "Acoustic Word Embedding System for Code-Switching Query-by-Example Spoken Term Detection," *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Hong Kong, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Ziwei Zhu et al., "Siamese Recurrent Auto-Encoder Representation for Query-by-Example Spoken Term Detection," *Interspeech*, pp. 102-106, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[35] Takumi Kurokawa, Atsuhiko Kai, and Hiroki Kondo, "Effects of End-to-end ASR and Score Fusion Model Learning for Improved Query-by-example Spoken Term Detection," *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Auckland, New Zealand, pp. 654-661, 2020. [Google Scholar] [Publisher Link]

[36] Jan Švec, Luboš Šmídl, and Jan Lehečka, "Transformer-based Encoder-Encoder Architecture for Spoken Term Detection." *Asian Conference on Pattern Recognition*, Kitakyushu, Japan, pp. 346-357, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] Chia-Wei Ao, and Hung-Yi Lee, "Query-by-Example Spoken Term Detection Using Attention-Based Multi-Hop Networks," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, pp. 6264-6268, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[38] Kun Zhang et al., "Query-by-Example Spoken Term Detection using Attentive Pooling Networks," *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Lanzhou, China, pp. 1267-1272, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[39] Maulik C. Madhavi, and Hemant A. Patil, "Vocal Tract Length Normalization using a Gaussian Mixture Model Framework for Query-by-Example Spoken Term Detection," *Computer Speech & Language*, vol. 58, pp. 175-202, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[40] Yaodong Zhang et al., "Resource Configurable Spoken Query Detection using Deep Boltzmann Machines," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, pp. 5161-5164, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[41] P. Sudhakar, K. Sreenivasa Rao, and Pabitra Mitra, "A Novel Zero-Resource Spoken Term Detection Using Affinity Kernel Propagation with Acoustic Feature Map," *SN Computer Science*, vol. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[42] Akanksha Singh, Vipul Arora, and Yi-Ping Phoebe Chen, "An Efficient TF-IDF based Query by Example Spoken Term Detection," *2024 IEEE Conference on Artificial Intelligence (CAI)*, Singapore, pp. 170-175, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[43] Prajyot Naik et al., "Kernel-based Matching and a Novel Training Approach for CNN-Based QbE-STD," *2020 International Conference on Signal Processing and Communications (SPCOM)*, Bangalore, India, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[44] 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. [CrossRef] [Google Scholar] [Publisher Link]

[45] Ravi Shankar, C.M. Vikram, and S.R.M. Prasanna, "Spoken Keyword Detection using Joint DTW-CNN," *Interspeech*, pp. 117-121, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[46] Yougen Yuan et al., "Learning Acoustic Word Embeddings with Temporal Context for Query-by-Example Speech Search," *arXiv Preprint*, pp. 97-101, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[47] Luis J. Rodriguez-Fuentes et al., "High-Performance Query-by-Example Spoken Term Detection on the SWS 2013 Evaluation," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, pp. 7819-7823, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[48] Maulik C. Madhavi, and Hemant A. Patil, "Partial Matching and Search Space Reduction for QbE-STD," *Computer Speech & Language*, vol. 45, pp. 58-82, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[49] Xavier Anguera, and Miquel Ferrarons, "Memory Efficient Subsequence DTW for Query-by-Example Spoken Term Detection," *2013 IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, USA, pp. 1-6, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[50] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard, "Sparse Subspace Modeling for Query by Example Spoken Term Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 6, pp. 1130-1143, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[51] R. Kishore Kumar, Lokendra Birla, and K. Sreenivasa Rao, "A Robust Unsupervised Pattern Discovery and Clustering of Speech Signals," *Pattern Recognition Letters*, vol. 116, pp. 254-261, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[52] Nadia Benati, and Halima Bahi, "Self-Supervised Spoken Term Detection for Query by Example," *International Information and Engineering Technology Association*, pp. 1175-1181, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[53] John S. Garofolo et al., "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *Linguistic Data Consortium*, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[54] MediaEval, The 2011 Spoken Web Search Task, 2011. [Online]. Available: http://www.multimediaeval.org/mediaeval2011/SWS2011/

[55] Florian Metze et al., "The Spoken Web Search Task at MediaEval 2011," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, pp. 5165-5168, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[56] MediaEval, The 2012 Spoken Web Search Task, 2012. [Online]. Available: http://www.multimediaeval.org/mediaeval2012/sws2012/

[57] Sri Harsha Dumpala et al., "Analysis of Constraints on Segmental DTW for the Task of Query-by-Example Spoken Term Detection," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[58] MediaEval, The 2013 Spoken Web Search Task, 2013. [Online]. Available: http://www.multimediaeval.org/mediaeval2013/sws2013/

[59] Xavier Anguera et al., "Query-by-Example Spoken Term Detection Evaluation on Low-Resource Languages," *Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-Resourced Languages*, Petersburg, Russia, pp. 24-31, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[60] MediaEval, The 2014 Query by Example Search on Speech (QUESST), 2014. [Online]. Available: http://www.multimediaeval.org/mediaeval2014/quesst2014/

[61] MediaEval, The 2015 Query by Example Search on Speech (QUESST), 2015. [Online]. Available: http://www.multimediaeval.org/mediaeval2015/quesst2015/

[62] H. Tulsiani, and P. Rao, "The IIT-B Query-by-Example System for MediaEval 2015," *MediaEval*, pp. 1-3, 2015. [Google Scholar] [Publisher Link]

[63] Jingyong Hou, "The NNI Query-by-Example System for MediaEval 2015," *MediaEval*, pp. 1-3, 2014. [Google Scholar] [Publisher Link]

[64] Tanja Schultz, "GlobalPhone: A Multilingual Speech and Text Database Developed at Karlsruhe University," *Proceedings of the 7th International Conference on Spoken Language Processing*, pp. 345-348, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[65] Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe, "GlobalPhone: A Multilingual Text and Speech Database in 20 Languages," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, pp. 8126-8130, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[66] I. McCowan et al., "The AMI Meeting Corpus," *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, pp. 137-140, 2005. [Google Scholar] [Publisher Link]

[67] AMI Corpus Overview, 2006. [Online]. Available: https://groups.inf.ed.ac.uk/ami/corpus/overview.shtml

[68] Lwazi English ASR Corpus. Lwazi, 2013. [Online]. Available: https://repo.sadilar.org/items/28063254-e197-40c7-a488-f904a68550a8

[69] Charl van Heerden, Neil Kleynhans, and Marelie Davel, "Improving the Lwazi ASR Baseline," *Interspeech*, pp. 3534-3538, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[70] James Glass et al., "Recent Progress in the MIT Spoken Lecture Processing Project," *Interspeech*, pp. 2553-2556, 2007. [Google Scholar] [Publisher Link]

[71] John J. Godfrey, and Edward Holliman, "Switchboard-1 Release 2 LDC97S62," *Linguistic Data Consortium*, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[72] A. Martin et al., "*The DET Curve in Assessment of Detection Task Performance*," Technical Report, Defense Technical Information Center, pp. 1895-1898, 1997. [Google Scholar] [Publisher Link]

[73] Gautam Mantena, and Kishore Prahallad, "Use of GPU and Feature Reduction for Fast Query-by-Example Spoken Term Detection," *Proceedings of the 11th International Conference on Natural Language Processing*, Goa, India, pp. 56-62, 2014. [Google Scholar] [Publisher Link]

[74] Jonathan G. Fiscus, Jerome G. Ajot, and John S. Garofolo, "Results of the 2006 Spoken Term Detection Evaluation," *Procedding of ACM Special Interest Group in Information Retrieval*, pp. 45-50, 2007. [Google Scholar] [Publisher Link]

[75] Open Keyword Search Evaluation, NIST, 2006. [Online]. Available: https://www.nist.gov/itl/iad/mig/open-keyword-search-evaluation

[76] Albayzin Evaluation, 2016. [Online]. Available: https://iberspeech2016.inesc-id.pt/index.php/albayzin-evaluation/

[77] Javier Tejedor et al., "The Multi-Domain International Search on Speech 2020 Albayzin Evaluation: Overview, Systems, Results, Discussion, and Post-Evaluation Analyses," *Applied Sciences*, vol. 11, no. 18, pp. 1-39, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[78] Javier Tejedor et al., "Albayzin 2018 Spoken Term Detection Evaluation: A Multi-Domain International Evaluation in Spanish," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, pp. 1-37, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[79] P. Sudhakar, K. Sreenivasa Rao, and Pabitra Mitra, "Unsupervised Spoken Term Discovery using Pseudo Lexical Induction," *International Journal of Speech Technology*, vol. 26, pp. 801-816, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[80] Pantid Chantangphol, Theerat Sakdejayont, and Tawunrat Chalothorn, "Enhancing Word Discrimination and Matching in Query-by-Example Spoken Term Detection with Acoustic Word Embeddings," *Proceedings of the 6th International Conference on Natural Language and Speech Processing*, pp. 293-302, 2023. [Google Scholar] [Publisher Link]

[81] Manisha Naik Gaonkar et al., "Exploring the Effectiveness of Feature Reduction and Kernel-Based Matching for Query-by- Example Spoken Term Detection Using CNN," *IEEE Access*, vol. 12, pp. 194462-194474, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[82] Manisha Naik Gaonkar, Veena Thenkanidiyoor, and Aroor Dinesh Dileep, "A Parallel Computing approach to CNN-based QbE-STD using Kernel-Based Matching," *Journal of Supercomputing*, vol. 81, 2024. [CrossRef] [Google Scholar] [Publisher Link]