IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comprehensive Survey And Implementation Framework For Text-Based Spoken Term Detection Using Terrier And Hiemstra Language Model

Vrushali Ravindra Deshpande [1] , Sushil Venkatesh Kulkarni[2] , Suryakant B Kendre[3]

[1] M.Tech. 3rd semester Student, Department of Computer Science And Information Technology, MBES College of Engineering Ambajogai, India

[2] Professor and Head , Department of Computer Science And Engineering, MBES College of Engineering Ambajogai, India

[3] Assistant Professor and Guide , Department of Computer Science And Engineering, MBES College of Engineering Ambajogai, India

Abstract: Text-Based Spoken Term Detection (STD) is the task of detecting and temporally localizing user-defined textual query terms directly from large collections of unstructured speech data. Unlike Automatic Speech Recognition (ASR), which focuses on generating complete transcriptions, STD concentrates on accurately identifying specific keywords with minimal false alarms and missed detections. Over the last two decades, STD has evolved from template-based methods and HMM-GMM frameworks to modern deep learning and transformer-based architectures. This paper presents a comprehensive survey of text-based STD systems covering literature evolution, system architecture, feature extraction, keyword modeling, ASR-based and acoustic-based techniques, deep learning advancements, similarity matching strategies, standard evaluation metrics, benchmarking tools, key challenges, and real-world applications. In addition, a complete implementation framework using the QUESST-2014 dataset with the Terrier Information Retrieval platform and the Hiemstra Language Model is presented, with speech transcription performed using the HappyScribe platform.

Keywords- Text-based Spoken term detection, Automatic Speech Recognition, Terrier Information Retrieval, Hiemstra language model

1. INTRODUCTION

The past two decades have witnessed an unprecedented growth in digitally recorded speech due to the rapid expansion of mobile devices, VoIP services, call centers, video conferencing platforms, podcasting, broadcast news, medical dictation systems, and surveillance infrastructures. Unlike structured text, speech data is inherently unstructured, continuous, and time-dependent, making large-scale indexing and retrieval a non-trivial problem. This challenge has given rise to the domain of Speech Information Retrieval (SIR), within which Spoken Term Detection (STD) plays a central role [1], [12].

STD aims to automatically detect whether a user-defined keyword occurs within a speech stream and to determine its temporal boundaries. Based on query formulation, STD systems are divided into query-by-example STD, where the query is an audio snippet, and text-based STD, where the query is provided as text.

Text-based STD is particularly attractive for large-scale multimedia search systems because it avoids storing and processing acoustic query templates and allows direct use of conventional information retrieval models [13], [30].

Unlike conventional Automatic Speech Recognition (ASR), which seeks to minimize Word Error Rate (WER), STD focuses on detection reliability and ranking performance. As a result, STD systems are evaluated using detection-centric metrics such as Precision, Recall, Mean Average Precision (MAP), and Actual Term Weighted Value (ATWV) rather than WER [1], [29]. This distinction allows STD to remain effective even when full transcription accuracy is poor.

Text-based STD has critical real-world applications. In military and intelligence surveillance, it enables continuous monitoring of intercepted communications for security-sensitive keywords [1], [19]. In broadcast media, STD supports automatic indexing and retrieval of news archives [13]. In call centers, it enables compliance monitoring, fraud detection, and customer sentiment analysis [19]. In healthcare, it supports search within medical dictation and clinical conversations. In forensic investigations, STD helps law enforcement agencies filter large speech datasets for evidence-related content [19].

Despite its importance, STD remains an extremely challenging research problem. Speech is highly variable due to speaker differences, accents, background noise, reverberation, speaking rate, and recording channel effects. Out-of-vocabulary (OOV) words further complicate detection for word-based ASR systems [6], [18]. Moreover, most of the world's languages lack sufficient annotated speech data, making low-resource STD a major open research challenge [14], [18].

Recent advances in deep learning and self-supervised learning have significantly improved robustness and generalization. Wav2Vec 2.0 demonstrated that high-quality speech representations can be learned from raw audio using large amounts of unlabeled data [4]. HuBERT further improved masked prediction for speech units [10], while Whisper showed strong multilingual generalization in real-world noisy conditions [11]. At the same time, open-source toolkits such as Kaldi [7], ESPnet, OpenKWS [25], and the Terrier IR platform [30] have accelerated practical deployment and reproducibility of STD systems.

This paper aims to provide:

1) A comprehensive historical and technical survey of text-based STD.
2) A system-level architectural breakdown of STD pipelines.
3) A critical review of feature extraction, keyword modeling, and matching strategies.
4) A benchmark-oriented discussion of evaluation metrics and tools.
5) A complete implementation framework using QUESST-2014 + HappyScribe + Terrier +Hiemstra_LM.

2. LITERATURE SURVEY

The development of Spoken Term Detection (STD) has progressed through multiple technological phases, each shaped by advancements in speech processing, statistical modeling, and machine learning. Early systems focused on direct acoustic template matching, whereas later approaches introduced probabilistic modeling, ASR-driven architectures, and modern deep learning and transformer-based methods. This section synthesizes key contributions that have influenced STD research and summarizes major milestones documented in existing literature

2.1 Template-Based and DTW-Based STD

The earliest STD systems were based on direct acoustic template matching using Dynamic Time Warping (DTW). In these systems, a fixed acoustic representation of a keyword was aligned against continuous speech to find minimum-distance matches. Although effective for small vocabularies, DTW-based systems were computationally expensive and highly sensitive to speaker and channel mismatch. These limitations rendered them impractical for large-scale databases [2], [19].

2.2 Posterior Probability-Based STD

A major milestone was the landmark work by **Mangu et al. (2000)** [2], which introduced posterior probability-based keyword detection. Instead of using raw spectral features, speech was represented as frame-level posterior probabilities over phoneme classes. Keyword detection was performed by correlating phoneme posterior sequences with query phoneme models. This approach dramatically improved robustness against noise and speaker variability and formed the conceptual foundation for modern probabilistic STD.

2.3 NIST STD Evaluations and ATWV

The **NIST Spoken Term Detection Evaluations (2006–2010)** were instrumental in formalizing STD as a benchmark research problem. **Fiscus et al. (2007)** [1] introduced standardized datasets, query lists, scoring protocols, and the **Actual Term Weighted Value (ATWV)** metric. ATWV explicitly models the trade-off between missed detections and false alarms and remains the primary performance metric in STD research today [29].

2.4 HMM-GMM and ASR-Based STD

Between 2008 and 2013, most STD systems were built on top of HMM-GMM based ASR. Speech was converted into word lattices or phoneme lattices, and keywords were searched within these structures [15], [23]. Although these systems enabled large-scale spoken document retrieval, their performance degraded significantly under high noise and low-resource conditions due to transcription errors [14], [18].

2.5 Deep Neural Network Based STD

The adoption of deep learning marked a turning point in STD. **Chen et al. (2014)** [3] showed that small-footprint DNNs outperform GMMs for keyword spotting. **Sainath et al. (2015)** [8] demonstrated that CNNs learn robust spectral features directly from log-Mel inputs. **Graves et al. (2013)** [9] and **Sak et al. (2015)** [27] showed that RNN and LSTM architectures are highly effective for continuous keyword spotting and streaming detection.

2.6 Transformer and Self-Supervised STD

Recent STD research is dominated by transformer-based and self-supervised speech models. **Baevski et al. (2020)** [4] introduced Wav2Vec 2.0, which enabled representation learning directly from raw speech using contrastive objectives. **Hsu et al. (2021)** [10] proposed HuBERT, which improved masked prediction over latent speech units. **Radford et al. (2022)** [11] demonstrated that Whisper generalizes robustly across languages and noise conditions using large-scale weak supervision. These models have made **zero-shot multilingual STD** practically viable.

3. ARCHITECTURE OF A TEXT-BASED SPOKEN TERM DETECTION SYSTEM

A text-based Spoken Term Detection (STD) system is a multi-stage signal processing and information retrieval pipeline designed to transform unstructured speech into searchable textual or probabilistic representations and perform efficient keyword search. Unlike standard ASR systems that emphasize complete transcription, STD architectures are optimized for **fast keyword detection, robustness to transcription**

errors, and scalable retrieval. Modern STD systems integrate concepts from digital signal processing, statistical speech recognition, probabilistic modeling, and information retrieval [1], [12], [13].

A generic text-based STD system can be decomposed into **three major subsystems**:

1. Speech Processing Subsystem
2. Text Query Processing Subsystem
3. Search, Ranking, and Decision Subsystem

These subsystems operate sequentially but are tightly coupled in performance.

3.1 Speech Input and Preprocessing Subsystem

The speech signal captured from real-world environments is inherently distorted by background noise, channel effects, reverberation, and microphone variability. Therefore, preprocessing is a mandatory first step in any STD architecture [5], [22].

Key preprocessing operations include:

i) Pre-emphasis filtering:

A high-pass filter is applied to boost high-frequency components of speech, compensating for spectral tilt introduced during speech production.

ii) Noise reduction:

Techniques such as spectral subtraction, Wiener filtering, and statistical noise modeling are employed to suppress stationary background noise. Robust noise handling is especially critical for surveillance and broadcast STD [12], [18].

iii) Silence and pause removal:

Voice Activity Detection (VAD) is applied to remove non-informative silence segments, thereby reducing computation during feature extraction and search.

iv) Amplitude normalization:

Signal energy normalization reduces inter-channel and inter-speaker loudness variations.

These preprocessing steps significantly improve downstream feature stability and ASR accuracy, which directly impacts STD performance [5], [22].

3.2 Feature Extraction Subsystem

Feature extraction transforms the preprocessed waveform into a compact parametric representation that preserves phonetic content while discarding irrelevant information. This step is the **foundation of all subsequent acoustic modeling** [5], [8].

The most widely used features in STD include:

i) Mel-Frequency Cepstral Coefficients (MFCC):

MFCCs approximate the nonlinear frequency resolution of the human auditory system and provide strong phonetic discrimination. Hermansky [5] established MFCCs as a standard feature for speech recognition and keyword detection.

ii) Perceptual Linear Prediction (PLP):

PLP integrates perceptual auditory models such as equal-loudness curves and intensity-to-loudness compression [5].

iii) Log-Mel Filterbank Energies:

These features preserve detailed spectral structure and are particularly suited for deep learning architectures such as CNNs and transformers [8], [4].

Feature extraction typically follows the pipeline:

Framing → Windowing → FFT → Mel Filterbanks → Log Compression → DCT (for MFCC)

Robust normalization techniques such as Cepstral Mean and Variance Normalization (CMVN) and Vocal Tract Length Normalization (VTLN) are frequently applied to reduce speaker and channel mismatches [22].

3.3 Acoustic Modeling Subsystem

The acoustic model maps feature vectors to linguistic units such as phonemes, subwords, or characters. The evolution of acoustic modeling mirrors the broader evolution of speech recognition technology [7], [12].

3.3.1 HMM-GMM Acoustic Models

Early STD systems used **Hidden Markov Models with Gaussian Mixture Models (HMM-GMM)** to model phoneme distributions. These models enabled lattice-based keyword search but were limited by:

i) Linear separability assumptions
ii) Poor generalization in noisy speech [15], [23]

3.3.2 DNN-HMM Hybrid Models

Deep Neural Networks replaced GMMs as emission probability estimators while retaining the HMM decoding framework. This hybrid approach significantly improved modeling of nonlinear acoustic variations and became the industry standard during 2013–2018 [3], [7], [8].

3.3.3 End-to-End and Transformer-Based Models

Recent STD systems increasingly rely on **end-to-end transformer architectures**, where raw acoustic features are directly mapped to subword or character sequences using self-attention mechanisms [4], [10], [11]. These models eliminate the need for handcrafted feature pipelines and provide:

i) Long-context modeling
ii) Multilingual generalization
iii) Robustness to domain mismatch

3.4 Text Query Processing and Keyword Representation

In text-based STD, the user supplies queries in textual form. These queries undergo **Grapheme-to-Phoneme (G2P) conversion** to generate phonemic representations, enabling detection of Out-of-Vocabulary (OOV) words that do not exist in the ASR dictionary [6], [18].

Query units may be represented as:

Word-level models: Direct string matching in transcripts
Phoneme-level models: Language-independent matching
Subword-level models: BPE tokens, syllables, triphones [4], [6]

Phoneme-level and subword-level modeling significantly improve STD robustness for multilingual and low-resource conditions.

3.5 Search and Matching Subsystem

This subsystem performs similarity matching between the keyword representation and speech-derived representations. Several matching paradigms exist [2], [23]:

i) Lattice-based search:

Keywords are searched in ASR-generated word or phoneme lattices.

ii) Posterior probability matching:

Keyword phoneme models are correlated with frame-level phoneme posteriorgrams, offering noise robustness [2], [23].

iii) Dynamic Time Warping (DTW):

Sequence alignment algorithm for acoustic matching, mainly used in query-by-example STD.

iv) Neural attention-based matching:

Modern architectures use attention mechanisms to perform soft alignment between query and speech representations [4], [10].

3.6 Decision and Scoring Subsystem

The final stage applies thresholding and ranking to determine whether a keyword exists in a given speech segment. Each detection is assigned:

i) A **confidence score**
ii) A **time boundary (start and end time)**

Ranking is then performed using probabilistic retrieval models such as BM25 or **Hiemstra Language Model (Hiemstra_LM)** when integrated with IR platforms like Terrier [30].

4. FEATURE EXTRACTION TECHNIQUES FOR TEXT-BASED SPOKEN TERM DETECTION

Feature extraction is the process of transforming a raw speech waveform into a compact, discriminative, and noise-robust parametric representation suitable for acoustic modeling and keyword detection. Since all subsequent stages of a Spoken Term Detection (STD) system operate on extracted features, the quality of these features directly determines detection accuracy, robustness to noise, and generalization across speakers and channels [5], [12].

An ideal feature set for STD must satisfy the following properties:

i) Strong **phonetic discrimination**
ii) Robustness to **background noise and reverberation**
iii) Invariance to **speaker and channel variations**
iv) Low **computational complexity** for large-scale deployment

Over the years, several feature representations have been proposed and evaluated for STD, among which **MFCC, PLP, Log-Mel filterbanks, and their robust variants** dominate modern systems.

4.1 Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) are the most widely used acoustic features in ASR and STD due to their strong perceptual relevance and compact representation. MFCCs are derived from the short-time power spectrum of speech and approximate the **non-linear frequency resolution of the human auditory system** using the Mel scale [5].

MFCC Extraction Pipeline

The standard MFCC extraction process consists of:

1. **Pre-emphasis:**
   Enhances high-frequency components to compensate for vocal tract spectral tilt.
2. **Framing and Windowing:**
   Speech is divided into 20–25 ms frames with 10 ms overlap and multiplied by a Hamming window.
3. **Fast Fourier Transform (FFT):**
   Converts time-domain frames into frequency-domain spectra.
4. **Mel Filterbank Integration:**
   Spectral energies are passed through triangular Mel-spaced filters.
5. **Logarithmic Compression:**
   Mimics the logarithmic sensitivity of human hearing.
6. **Discrete Cosine Transform (DCT):**
   Produces de-correlated cepstral coefficients.

Why MFCC is Dominant in STD?

Hermansky[5] established MFCCs as the standard feature for speech recognition. Subsequent STD studies confirm that MFCCs:

a) Offer **strong phoneme discrimination**
b) Are **computationally efficient**
c) Perform reliably in **clean and moderately noisy conditions** [12], [23]

Limitations of MFCC

Despite their popularity, MFCCs suffer from:

i) Sensitivity to **additive noise**
ii) Performance degradation under **reverberation**
iii) Vulnerability to **channel mismatch**

These weaknesses motivated the development of perceptually enhanced and robust features.

4.2 Perceptual Linear Prediction (PLP)

Perceptual Linear Prediction (PLP) integrates models of the human auditory system more explicitly than MFCC. Introduced by Hermansky [5], PLP incorporates:

i. Critical band (Bark-scale) frequency warping
ii. Equal-loudness pre-emphasis
iii. Intensity-loudness power law compression

PLP features often demonstrate performance comparable to MFCCs in clean speech but show **better robustness in some noisy and mismatched conditions**. Several STD studies report improved phoneme classification using PLP over MFCC in low-SNR environments [12], [19].

4.3 Log-Mel Filterbank Energies

Log-Mel Filterbank Energies (Log-Mel) retain the Mel-scaled spectral energies **without the DCT step**. While MFCCs aim for decorrelation, Log-Mel features preserve spatial locality in the frequency domain.

Importance in Deep Learning-Based STD

CNN and transformer-based STD systems typically use Log-Mel features as direct inputs because:

a) CNNs exploit **local spectral correlations**
b) Transformers benefit from **preserved temporal-frequency structure**
c) Avoiding DCT prevents loss of discriminative spectral cues [8], [4]

Sainath et al. [8] and later transformer-based works [4], [10] demonstrated that Log-Mel features significantly outperform MFCCs when used with deep neural architectures.

4.4 Robust Feature Variants for Noisy STD

Since STD systems frequently operate under adverse conditions (surveillance, broadcast speech, outdoor recordings), robust feature extraction is essential. Several enhancements are commonly applied:

4.4.1 RASTA-PLP

RelAtive SpecTrAl (RASTA) filtering suppresses slowly varying spectral components such as channel effects and emphasizes modulation frequencies important for speech. RASTA-PLP is widely used in noisy STD and low-resource ASR [22].

4.4.2 Power-Normalized Cepstral Coefficients (PNCC)

PNCC features improve robustness by:

a. Using medium-time power normalization
b. Suppressing background noise
c. Modeling auditory masking effects more accurately than MFCC

PNCCs are particularly effective in highly non-stationary noise environments [22].

4.4.3 Cepstral Mean and Variance Normalization (CMVN)

CMVN normalizes cepstral features to zero mean and unit variance per utterance or per speaker, reducing:

a. Channel distortion
b. Microphone variability
c. Inter-speaker loudness differences

CMVN is considered **mandatory preprocessing** in modern STD systems [7], [22].

4.4.4 Vocal Tract Length Normalization (VTLN)

VTLN compensates for anatomical differences in human vocal tracts by frequency warping. It improves speaker normalization, particularly in multi-speaker STD tasks [22].

4.5 Embedded Feature Learning Using Deep Models

Unlike classical STD systems that rely on handcrafted features, modern deep learning systems increasingly perform **implicit feature learning** directly from:

i. Raw waveform samples
ii. Log-Mel filterbank tensors

CNN-Based Feature Learning

CNNs automatically learn:

i) Noise-robust spectral filters
ii) Speaker-invariant acoustic patterns
This significantly reduces reliance on handcrafted features [8].

Self-Supervised Feature Learning

Self-supervised models such as **Wav2Vec 2.0** [4] and **HuBERT** [10] learn high-level phonetic and semantic representations from unlabeled speech. These embeddings:

1. Serve as universal speech features
2. Transfer well to multilingual STD
3. Perform strongly in low-resource settings

4.6 Feature Selection for Text-Based STD Using ASR + Terrier

In ASR-driven text-based STD pipelines (including our **HappyScribe → Terrier → Hiemstra_LM** framework), feature extraction directly affects:

i) ASR transcription quality
ii) Word error rate
iii) Keyword recall and false alarm rate

Although the feature extraction step is hidden inside HappyScribe's proprietary ASR engine, its internal architecture is expected to rely on **Log-Mel + Transformer-based representations**, consistent with modern cloud ASR platforms [11]. Any degradation in features propagates as transcription errors, which subsequently impacts Terrier-based retrieval and MAP scores.

5. Keyword Modeling Strategies for Text-Based Spoken Term Detection

Keyword modeling defines how a textual query term is represented for matching against speech-derived representations in a Spoken Term Detection (STD) system. In text-based STD, the keyword is provided as a text string, but it must be transformed into a representation that is compatible with the underlying search space, which may consist of word sequences, phoneme sequences, subword units, or probabilistic posterior distributions**.** The choice of keyword modeling strategy has a direct and often dominant effect on Out-of-Vocabulary (OOV) detection capability, multilingual scalability, false alarm rate, and detection robustness [2], [6], [12].

An effective keyword model must satisfy the following requirements:

i. Robust to pronunciation variation
ii. Capable of handling OOV terms
iii. Compatible with noisy ASR outputs
iv. Scalable to large query vocabularies
v. Language-independent for multilingual STD

Based on these criteria, keyword modeling strategies are broadly categorized into:

1. Word-Level Keyword Modeling
2. Phoneme-Level Keyword Modeling
3. Subword and Unit-Based Keyword Modeling
4. Probabilistic Keyword Modeling

5.1 Word-Level Keyword Modeling

In word-level keyword modeling, the query is represented directly as a sequence of words and matched against **ASR-generated word transcripts or word lattices**. This is the most straightforward approach and is widely used in industrial STD pipelines due to its simplicity and computational efficiency [1], [13].

Operational Principle

1. Speech is fully transcribed using ASR.
2. The keyword is matched as an exact or approximate string within:
   i) ASR transcripts
   ii) Word lattices
   iii) Confusion networks

Advantages

1. Simple implementation
2. Direct compatibility with text-based IR systems (e.g., Terrier)
3. Low computational overhead

Limitations

1. Complete failure for OOV words
2. Highly sensitive to ASR substitution and deletion errors
3. Performs poorly in noisy and low-resource environments [14], [18]

In our implementation pipeline (**HappyScribe → Terrier → Hiemstra_LM**), keyword modeling is primarily **word-based**, since Terrier operates directly on word tokens. This makes our system computationally efficient but also **fundamentally bounded by transcription accuracy**.

5.2 Phoneme-Level Keyword Modeling

To overcome the OOV problem in word-level modeling, phoneme-level keyword modeling represents the keyword as a sequence of phonemes obtained through Grapheme-to-Phoneme (G2P) conversion [6], [18].

Operational Principle

1. Text keyword → G2P → phoneme sequence
2. Speech → phoneme posterior probabilities (from ASR or acoustic model)
3. Keyword detection is performed by matching phoneme sequences using:
    i. Posteriorgram correlation
    ii. DTW
    iii. Lattice-based phoneme search [2], [23]

Advantages

1. Enables **OOV detection**
2. Language-independent representation
3. More robust to **pronunciation variations**

Limitations

1. Higher computational cost than word-based STD
2. Sensitive to **phoneme recognition errors**
3. Requires a reliable **phonetic alphabet and G2P rules** for each language

Sun et al. [6] demonstrated that phoneme-based STD significantly outperforms word-based systems in **low-resource languages** where ASR vocabularies are limited.

5.3 Subword and Unit-Based Keyword Modeling

Subword modeling represents keywords using intermediate units that lie between words and phonemes. These units include:

a. Syllables
b. Triphones
c. Morphemes
d. Byte Pair Encoding (BPE) subwords **[4]**

Motivation

1. Word-level models are too rigid
2. Phoneme-level models are too granular
3. Subwords achieve a balance between discrimination and generalization

Applications in Modern STD

Transformer-based STD systems often tokenize speech into:

1. BPE units
2. Sentence Piece units which are then used for both transcription and keyword matching [4], [10], [11].

Advantages

1. Strong handling of OOV and rare words
2. Better modeling of co-articulation than phonemes
3. Improved cross-lingual transferability

Limitations

1. Model training is more complex
2. Unit inventory design impacts performance

Subword modeling is currently the dominant approach in multilingual transformer-based STD systems**.**

5.4 Probabilistic Keyword Modeling

Instead of representing keywords as deterministic symbol sequences, probabilistic keyword modeling represents them as **probability distributions over phoneme or subword states**. This approach directly models pronunciation uncertainty and ASR ambiguity [2], [23].

Probabilistic Model Structure

A keyword is represented as:

$$K = \{ P(p_1|t_1), P(p_2|t_2), \ldots, P(p_n|t_n) \}$$

where $P(p_i|t_i)$ represents the probability of phoneme p_i at time t_i.

Advantages

1. Tolerant to acoustic variation and pronunciation errors
2. Reduces false rejections
3. Performs well in low SNR environments

Karafiát et al. [23] showed that posterior-based probabilistic keyword models significantly outperform hard phoneme-string matching in noisy speech.

5.5 Multilingual Keyword Modeling

Multilingual STD introduces additional challenges:

1. Different phoneme inventories
2. Script variations
3. Grapheme-to-phoneme ambiguity

Modern multilingual STD systems rely on:

1) Shared multilingual phoneme sets
2) Universal BPE token vocabularies
3) Language-agnostic speech embeddings [4], [10], [18]

Rousseau et al. [26] demonstrated that multilingual keyword modeling significantly improves detection performance in cross-lingual STD tasks.

5.6 Keyword Modeling in ASR + Terrier-Based Text STD

In our implementation framework**,** keyword modeling is performed at the word level, because:

i. HappyScribe provides word-level transcriptions
ii. Terrier indexes word tokens
iii. Hiemstra_LM computes term probabilities for words

Thus, our pipeline operates as:

Text Keyword → Word Token → Terrier Query → Hiemstra_LM Ranking

Consequences of This Design

[1] Excellent scalability
[2] Direct MAP optimization
[3] Low computation cost
[4] No direct OOV handling
[5] Performance limited by ASR word accuracy

This makes our system ideal for:

1) Medium-to-high resource languages
2) Clean-to-moderate noise environments
3) Large-scale spoken document retrieval

Future extensions can integrate phoneme or subword keyword modeling upstream of Terrier to address the OOV limitation.

5.7 Comparative Summary of Keyword Modeling Strategies

Table 1 shows the comparative summary OF KEYWORD MODELING STRATEGIES:

Table 1 KEYWORD MODELING STRATEGIES

Modeling Level	OOV Support	Noise Robustness	Complexity	Typical Use Case
Word-Level	No	Medium	Low	Large-scale IR-based STD
Phoneme-Level	Yes	High	High	Low-resource and noisy STD
Subword-Level	Yes	High	Medium	Multilingual transformer STD
Probabilistic	Yes	Very High	Very High	Surveillance-grade STD

6. DEEP LEARNING IN TEXT-BASED SPOKEN TERM DETECTION

Deep learning has fundamentally transformed the field of Spoken Term Detection (STD) by enabling automatic learning of robust, hierarchical speech representations directly from data. Unlike traditional HMM-GMM based systems that relied heavily on handcrafted features and strong statistical assumptions, deep neural architectures learn non-linear mappings that significantly improve robustness to noise, speaker variability, and pronunciation diversity [3], [8], [9], [12]. Today, almost all state-of-the-art STD systems are based on deep learning, particularly CNNs, RNNs/LSTMs, Transformers, and self-supervised speech models**.**

The application of deep learning to STD can be broadly categorized into four major paradigms:

1. CNN-Based STD
2. RNN/LSTM-Based STD
3. Transformer-Based STD
4. Self-Supervised Representation Learning for STD

6.1 CNN-Based Spoken Term Detection

Convolutional Neural Networks (CNNs) were the first deep learning architectures to demonstrate significant improvements in keyword spotting and STD. CNNs operate directly on time–frequency representations such as Log-Mel filterbanks or spectrograms and learn local spectral filters that are robust to noise and speaker variation [8].

Working Principle

1. Input: Log-Mel or spectrogram features
2. Convolutional layers learn frequency-local patterns
3. Pooling layers provide translation invariance
4. Fully connected layers perform keyword classification

Key Contributions from Literature

1. Sainath et al. (2015) **[**8] demonstrated that CNNs outperform traditional GMM-HMM acoustic models for large-scale speech recognition and keyword spotting.
2. CNNs were found to be particularly effective in handling additive noise and channel distortions common in real-world STD tasks.

Advantages

1. Strong noise robustness
2. Efficient parallel computation on GPUs
3. Low-latency inference suitable for embedded STD

Limitations

1. Limited ability to model long-range temporal dependencies
2. Requires large labeled datasets for optimal performance

CNN-based STD systems are still widely used in wake-word detection and embedded keyword spotting**.**

6.2 RNN and LSTM-Based Spoken Term Detection

While CNNs capture local spectral patterns, Recurrent Neural Networks (RNNs) and particularly Long Short-Term Memory (LSTM) networks are designed to model temporal dynamics in speech signals, which is crucial for STD [9], [27].

Working Principle

1. Speech frames are processed sequentially
2. Hidden state carries contextual memory across time
3. LSTM gates regulate information flow across long durations

Key Contributions from Literature

1. Graves et al. (2013) [9] demonstrated that deep RNNs vastly outperform HMM-based acoustic models in continuous speech recognition.
2. Sak et al. (2015) [27] showed that LSTM networks are highly effective for real-time keyword spotting and streaming STD.

Advantages

1. Excellent modeling of long-term phonetic dependencies
2. Ideal for continuous and streaming STD
3. Lower false alarm rates compared to CNN-only systems

Limitations

1. Sequential nature limits parallelization
2. Higher computational cost during training

RNN/LSTM architectures remain a cornerstone of real-time STD systems used in voice-activated assistants and call-center analytics.

6.3 Transformer-Based Spoken Term Detection

Transformers represent the current state of the art in speech and language processing. Unlike CNNs and RNNs, transformers rely entirely on self-attention mechanisms to model global contextual relationships [4], [10], [11].

Core Idea

1. Every frame attends to all other frames
2. Global context is modeled in a single layer
3. No recurrence or convolution required

Major Research Milestones

1. Wav2Vec 2.0 (Baevski et al., 2020) **[**4] introduced a transformer encoder trained using contrastive self-supervised learning directly on raw speech.
2. HuBERT (Hsu et al., 2021) [10] improved masked speech unit prediction using iterative clustering.
3. Whisper (Radford et al., 2022) [11] demonstrated large-scale multilingual transformer training with weak supervision, showing exceptional generalization to noisy and low-resource speech.

Advantages

1. Superior long-context modeling
2. Strong multilingual and cross-domain generalization
3. Enables zero-shot STD (querying languages not seen during training)

Limitations

1. High memory and computation requirements
2. Not ideal for low-power embedded systems

Transformer-based STD now dominates research-grade and industrial cloud-based speech retrieval systems**.**

6.4 Self-Supervised Learning for Spoken Term Detection

Self-supervised learning has revolutionized STD by removing the dependency on large labeled speech corpora**.** In self-supervised models, the network learns representations by predicting missing or masked portions of the input without manual phoneme or word annotations [4], [10], [18].

Key Self-Supervised Models

1. Wav2Vec 2.0 [4]
2. HuBERT [10]
3. CPC (Contrastive Predictive Coding)
4. Whisper (weakly supervised variant) [11]

Impact on STD

1. Enables robust STD in low-resource languages
2. Supports zero-shot keyword detection
3. Produces speech embeddings usable for both ASR and STD

Thomas et al. [18] showed that zero-resource speech processing using unsupervised and self-supervised models enables effective phonetic discovery and keyword detection even without labeled data.

6.5 Deep Learning for Acoustic vs. Text-Based STD

Table 2 shows the DEEP LEARNING FOR ACOUSTIC VS. TEXT-BASED STD.

Table 2 ACOUSTIC VS. TEXT-BASED STD

Aspect	Acoustic STD	Text-Based STD
Input to DL Model	Acoustic features	ASR transcripts or embeddings
OOV Handling	Strong	Weak (word-based)
Noise Robustness	High	Moderate
System Complexity	High	Low
Scalability	Medium	Very High

Deep learning is used:

I. In acoustic STD for direct query-to-speech matching
II. In text-based STD indirectly through ASR transcription (as in our HappyScribe pipeline)

Our system leverages deep learning implicitly because HappyScribe uses transformer-based ASR internally, even though the model itself is not directly accessible.

6.6 Role of Deep Learning in Our Implementation Framework

In our QUESST-2014 + HappyScribe + Terrier + Hiemstra_LM pipeline:

[1] Deep learning is embedded inside HappyScribe's proprietary ASR

[2] The ASR likely uses:

i. Log-Mel + Transformer encoders

ii. Self-supervised pretraining similar to Whisper or Wav2Vec 2.0

[3] Output quality directly affects:

i. Word accuracy

ii. Keyword recall

iii. MAP computed by Terrier

Thus, although our retrieval layer is classical IR-based, the front-end transcription is entirely deep-learning driven, making your overall system a hybrid Deep Learning + Probabilistic IR STD framework**.**

6.7 Current Research Trends in Deep Learning-Based STD

Modern STD research emphasizes:

1. Zero-shot STD using multilingual transformers [4], [11]
2. Cross-lingual transfer learning [26]
3. Emotion-aware keyword detection
4. Multimodal STD combining audio and video
5. Edge-AI STD using compressed neural networks
6. Federated learning for privacy-preserving speech retrieval

These trends define the future direction of STD system development.

7. SIMILARITY MATCHING TECHNIQUES IN TEXT-BASED SPOKEN TERM DETECTION

Similarity matching is the fundamental operation that determines whether a given keyword representation matches a segment of speech. After feature extraction, acoustic modeling, and keyword representation, the STD system must compute a **similarity score or likelihood function** that quantifies how well the query matches the speech signal. This stage directly controls the **trade-off between miss rate and false alarm rate**, and therefore dominates overall system performance [1], [2], [12].

Similarity matching techniques in STD can be broadly classified into:

1. Dynamic Time Warping (DTW)-Based Matching
2. Posterior Probability-Based Matching
3. Embedding-Based Similarity Matching
4. Neural Attention-Based Matching
5. Text-Based Probabilistic Retrieval Matching (IR-Based STD)

Each class is optimized for different deployment constraints such as low-resource conditions, large-scale retrieval, or real-time operation.

7.1 Dynamic Time Warping (DTW)-Based Matching

Dynamic Time Warping (DTW) is one of the earliest and most widely used sequence alignment algorithms in STD. It computes the minimum-distance alignment between two time-dependent sequences of unequal length, typically between:

a) A keyword acoustic template

b) A segment of continuous speech

Mathematical Principle

Given two sequences

$$X=\{x_1,x_2,\ldots,x_T\}, \quad Y=\{y_1,y_2,\ldots,y_S\}$$

DTW finds a warping path W that minimizes:

$$DTW(X,Y) = \min_{W} \sum_{(i,j)\in W} d(xi\,.\,yi) \qquad (7.1)$$

where d(·) is usually Euclidean or cosine distance.

Advantages

1. Flexible time alignment
2. No need for phonetic or textual modeling
3. Effective for query-by-example STD

Limitations

1. Computational complexity O(TS)
2. Poor scalability for large databases
3. Sensitive to noise and speaker mismatch

DTW was widely used in early template-based STD systems but is now limited to small-scale or biometric-style keyword spotting [2], [19].

7.2 Posterior Probability-Based Similarity Matching

Posterior probability-based matching represents speech as frame-level phoneme posterior probability vectors instead of raw spectral features. This was a major breakthrough introduced by Mangu et al. (2000) [2].

Working Principle

1. Acoustic model produces phoneme posteriors for each frame:
$$P(p_k|x_t)$$
2. The keyword is represented as a phoneme sequence.
3. Similarity is computed by correlating keyword phoneme models with posterior probability trajectories.

Scoring Method

The keyword likelihood over a speech segment is computed as:

$$Score(K,X) = \prod_{t=1}^{T} \sum_{k=1}^{K} P(pk|xt).P(pk|K) \qquad (7.2)$$

Advantages

1. Highly robust to noise
2. Tolerant to pronunciation variation
3. Works well in low-resource languages

Limitations

1. Requires a reliable phoneme recognizer
2. Computationally more expensive than word-based matching

Karafiát et al. [23] showed that posterior-based STD significantly outperforms lattice-only matching in noisy conditions.

7.3 Embedding-Based Similarity Matching

Modern deep learning-based STD systems often represent both speech segments and keywords as fixed-length embeddings in a shared latent space**.** Similarity is then computed using vector distance metrics.

Common Similarity Measures

1. Cosine similarity

$$\boldsymbol{Cos(\theta) = \frac{A.B}{||A|||B||}} \qquad (7.3)$$

2. Euclidean distance
3. Dot-product attention scores

Advantages

1. Extremely fast retrieval using vector indexing
2. Scales well to very large speech archives
3. Enables approximate nearest-neighbor search

Limitations

1. Requires deep embedding training
2. Detection accuracy depends on representation quality

Embedding-based STD is widely used in transformer and self-supervised architectures [4], [10], [11].

7.4 Neural Attention-Based Similarity Matching

Neural attention replaces explicit DTW alignment with **soft, learnable alignment mechanisms**. This approach is now standard in transformer-based STD systems.

Key Concept

Each speech frame attends to each keyword unit using:

$$\mathbf{Attention(Q, K, V) = softmax}\left(\frac{\mathbf{QK^T}}{\sqrt{\mathbf{dk}}}\right)\mathbf{V} \qquad \mathbf{(7.4)}$$

Advantages

1. Learns optimal alignment automatically
2. Eliminates explicit DTW computation
3. Robust to speaking rate variation

Limitations

1. High computation and memory cost

2. Requires large training data

Attention-based matching is central to modern zero-shot and multilingual STD systems [4], [10].

7.5 Probabilistic Text-Based Similarity Matching Using IR Models

In text-based STD, similarity matching is performed at the textual IR level rather than the acoustic level. This is the class to which your Terrier + Hiemstra_LM framework belongs.

7.5.1 Basic Principle

Speech is first converted into text using ASR. Keyword similarity is computed as a probabilistic relevance score between:

a) Query **q**
b) Transcribed document **d**

$$\textbf{Score(d, q) = f(P(t|d),P(t|C))} \qquad \textbf{(7.5)}$$

7.5.2 Hiemstra Language Model for Similarity Matching

The Hiemstra Language Model (Hiemstra_LM) is a probabilistic retrieval model that computes the likelihood of observing a query from a document language model with background smoothing [30].

$$\boldsymbol{Score(d, q) = \sum_{t \in q} \log(\lambda P(t|d) + (1-\lambda)P(t|C))} \qquad (7.6)$$

Where:

P(t|d) = f(t,d) / |d|
P(t|C)=f(t,C) / |C|
λ is the smoothing parameter.

Significance for STD

1. Robust to ASR word deletions and substitutions
2. Naturally handles partial keyword matches
3. Directly optimizes ranking quality (MAP)

Our system uses:

trec.model = Hiemstra_LM
hiemstra.lambda = 0.25

Thus, similarity matching in your STD framework is probabilistic, corpus-smoothed, and optimized for large-scale spoken document ranking**.**

7.6 Thresholding , Decision Rules, and False Alarm Control

After similarity scoring, decisions are made using:

i. Fixed thresholds
ii. Score normalization
iii. False alarm penalties

NIST's **ATWV metric** explicitly models the cost of false alarms and missed detections, forcing systems to balance sensitivity and specificity [1], [29].

7.7 Comparative Summary of Similarity Matching Techniques

Table 3 shows the comparative summary of Similarity Matching Techniques

Table 3 SIMILARITY MATCHING TECHNIQUES

Matching Type	Domain	Robustness	Scalability	Typical Use
DTW	Acoustic	Medium	Very Low	Query-by-example
Posterior-Based	Acoustic	Very High	Medium	Low-resource STD
Embedding-Based	Latent Space	Very High	Very High	Large-scale deep STD
Attention-Based	Neural	Very High	Medium	Transformer STD
IR-Based (Hiemstra)	Text	Medium	**Very High**	Spoken document retrieval

Role of Similarity Matching in Our Implementation

In our system:
HappyScribe ASR → Text → Terrier Inverted Index → Hiemstra_LM → Similarity Scores → MAP

Thus:

a) We do not match acoustic patterns directly
b) We perform probabilistic text similarity matching
c) Our performance depends on:
   i. Transcription quality
   ii. Term probability estimation
   iii. Smoothing parameter λ

This makes our system:

1. Highly scalable
2. Retrieval-optimized
3. Suitable for large speech archives
4. Sensitive to ASR errors
5. Limited for OOV detection

8. EVALUATION METRICS AND BENCHMARKING FRAMEWORK FOR SPOKEN TERM DETECTION

Evaluation in Spoken Term Detection (STD) is fundamentally different from conventional Automatic Speech Recognition (ASR) evaluation. While ASR focuses on minimizing Word Error Rate (WER), STD is a detection and ranking task, where the objective is to reliably identify the presence of a keyword in a large speech archive while minimizing missed detections and false alarms. Consequently, STD systems are evaluated using detection-oriented and ranking-oriented metrics, along with standardized benchmarking protocols [1], [12], [29].

A well-designed evaluation framework for STD must satisfy the following properties:

1. Measure both detection accuracy and ranking quality
2. Penalize false alarms and missed detections explicitly
3. Be query-independent and corpus-scalable

4. Support cross-system comparison
5. Be compatible with large speech archives

The most widely used evaluation metrics in STD research are:

1. Precision
2. Recall
3. Mean Average Precision (MAP)
4. Actual Term Weighted Value (ATWV)

8.1 Precision and Recall

Precision and Recall are fundamental information retrieval metrics adapted to STD for measuring detection reliability [12], [13].

Precision

Precision measures the proportion of detected keyword instances that are actually correct:

$$\textbf{Precision} = \frac{\textbf{True Positives(TP)}}{\textbf{TP} + \textbf{False positive (FP)}}$$

High Precision → Low false alarm rate
Low Precision → Many incorrect keyword detections

In surveillance and forensic applications, high Precision is often more critical than high Recall, since false alarms can overwhelm analysts [19].

Recall

Recall measures the proportion of actual keyword occurrences that are successfully detected:

$$\textbf{Recall} = \frac{\textbf{True Positives(TP)}}{\textbf{TP+False Negatives(FN)}}$$

High Recall → Fewer missed detections
Low Recall → Important keyword occurrences are lost

In intelligence and security monitoring, high Recall is mandatory to avoid missing critical events [1], [29].

Precision–Recall Trade-Off

There is an inherent trade-off between Precision and Recall controlled by:

[1] Detection threshold
[2] Language-model smoothing parameter
[3] ASR confidence calibration

Optimizing this trade-off is one of the central research problems in STD system design [12].

8.2 Mean Average Precision (MAP)

Mean Average Precision (MAP) is the primary ranking-based evaluation metric used in spoken document retrieval and text-based STD systems [13], [30]. It evaluates how well the system ranks relevant speech documents for a given keyword query.

Average Precision (AP) for a Single Query

$$AP = \frac{1}{R} \sum_{k=1}^{N} (P(k) \cdot rel(k)) \qquad (8.1)$$

where:

R = total number of relevant documents
P(k) = precision at rank k
rel(k) = relevance indicator (1 if relevant, 0 otherwise)

Mean Average Precision

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} (AP_q) \qquad (8.2)$$

where **Q** is the total number of queries.

Why MAP is Ideal for our System?

Our pipeline:

HappyScribe → Terrier → Hiemstra_LM → Ranked Speech Documents

is a pure ranking-oriented STD framework**,** making MAP the most appropriate primary metric**.** MAP directly reflects:

[1] Effectiveness of Hiemstra_LM
[2] Impact of ASR transcription quality
[3] Retrieval accuracy of Terrier indexing

MAP is also the official performance metric in TREC and QUESST-style spoken document retrieval tasks [13], [30].

8.3 Actual Term Weighted Value (ATWV)

The Actual Term Weighted Value (ATWV) is the official metric defined by NIST for STD evaluation [1], [29]. It explicitly balances the trade-off between missed detections and false alarms using a cost-sensitive formulation.

$$ATWV = 1 - (P_{Miss} + \beta \cdot P_{FA}) \qquad (8.3)$$

Where:

P_{Miss} = Probability of miss
P_{FA} = Probability of false alarm
β = false-alarm penalty factor

Key Properties of ATWV

1. Strongly penalizes false alarms
2. Reflects real-world surveillance priorities
3. Supports system-level tuning
4. Used exclusively in NIST STD evaluations

While our current implementation primarily reports MAP**,** ATWV remains important for:

1. Comparative benchmarking with global STD systems
2. Evaluating threshold-based detection behavior

8.4 Detection Error Tradeoff (DET) Curves

Detection Error Tradeoff (DET) curves visualize the relationship between:

1. False alarm probability
2. Miss probability
   on a normalized Gaussian scale [1].

DET curves are especially useful for:

a) Comparing multiple STD systems
b) Selecting optimal operating thresholds
c) Visualizing system robustness under different cost constraints

DET analysis is standard in NIST STD evaluations and surveillance-grade STD research.

8.5 Standard Benchmark Datasets for STD

8.5.1 NIST STD Corpora

The NIST STD corpora (2006–2010) established:

1. Standard query lists
2. Ground-truth annotations
3. Official ATWV scoring rules [1], [29]

8.5.2 QUESST-2014 Dataset

QUESST-2014 is a multilingual STD benchmark consisting of:

1. Speech audio files
2. Textual keyword queries
3. Relevance judgments

It is specifically designed for:

a) Low-resource STD
b) Multilingual keyword search
c) Real-world noise conditions

Our implementation is based on **QUESST-2014**, making it directly comparable with international STD research.

8.5.3 TREC Spoken Document Retrieval Track

The TREC SDR track focuses on:

1. Spoken news archives
2. ASR-based retrieval
3. MAP-based ranking evaluation [13]

Our **Terrier + Hiemstra_LM + MAP** framework is directly aligned with TREC-style evaluation.

8.6 Benchmarking Toolkits for STD Evaluation

8.6.1 Terrier Information Retrieval Platform

Terrier is a widely used open-source IR system for:

1. Document indexing
2. Probabilistic retrieval models
3. MAP-based evaluation [30]

Terrier natively supports:

a) BM25
b) TF-IDF
c) Hiemstra_LM
d) TREC-style evaluation scripts

This makes Terrier ideal for ASR-driven text-based STD benchmarking**.**

8.6.2 Kaldi and OpenKWS

a) Kaldi supports phoneme lattices, posterior extraction, and acoustic STD [7]
b) OpenKWS provides a complete toolkit for acoustic keyword spotting [25]

These tools are used in acoustic-level STD benchmarking, complementary to your text-based IR pipeline.

8.6.3 NIST STD Evaluation Toolkit

The official NIST scoring toolkit computes:

a) ATWV
b) P(Miss)
c) P(FA)
d) DET curves [29]

This toolkit ensures internationally standardized STD evaluation**.**

8.7 Evaluation Methodology in Your Implementation

Our evaluation pipeline is formally defined as:

QUESST-2014 Ground Truth
↓
HappyScribe Transcripts → Terrier Index → Hiemstra_LM Ranking
↓
Precision – Recall – MAP

Evaluation Steps

1. Generate transcriptions using HappyScribe
2. Index transcripts using Terrier
3. Submit keyword queries
4. Retrieve ranked speech documents
5. Compare retrieved results with QUESST-2014 relevance judgments
6. Compute Precision, Recall, and MAP

This methodology is:

i. Reproducible
ii. Standard-compliant
iii. Aligned with TREC and QUESST evaluation protocols
iv. Suitable for large speech archives

8.8 Limitations of Current Evaluation Strategy

Although MAP-based evaluation is ideal for ranking-oriented STD, it does not explicitly penalize false alarms at the score level**.** Also:

1. ATWV is not directly computed in your current pipeline
2. Temporal localization accuracy is not explicitly evaluated
3. ASR transcription uncertainty is not explicitly modeled

These limitations define important directions for future experimental extensions**.**

9. CHALLENGES AND OPEN RESEARCH PROBLEMS IN TEXT-BASED SPOKEN TERM DETECTION

Despite major advances in acoustic modeling, self-supervised learning, multilingual ASR, and transformer architectures, Spoken Term Detection (STD) remains far from solved**.** The problem is inherently difficult due to the nature of human speech—its variability, ambiguity, and sensitivity to real-world conditions. Existing STD systems, including ASR-driven text-based retrieval frameworks like the one proposed in this study (HappyScribe → Terrier → Hiemstra_LM), still face significant technical, linguistic, and operational challenges.

These challenges define active research frontiers and motivate new algorithmic directions in both academic and industrial STD development.

9.1 High Sensitivity to ASR Errors in Text-Based STD

Text-based STD inherits all weaknesses of its underlying ASR system. If the ASR fails to correctly recognize:

1. highly accented speech,
2. code-mixed language,
3. rare proper nouns,
4. OOV terms, or
5. domain-specific jargon,

then no amount of downstream IR ranking can recover the lost keyword [12], [18].

This issue is especially severe in:

[1] Noisy speech
[2] Low-resource languages
[3] Informal conversational speech
[4] Whispered or emotional speech

In our pipeline, this means HappyScribe's transcription errors directly propagate into Terrier and degrade Precision, Recall, and MAP**.**

9.2 Out-of-Vocabulary (OOV) Word Detection

OOV keyword detection remains a fundamental unsolved problem in text-based STD. Word-based systems (including yours) fail completely when:

a) the keyword is not present in the ASR dictionary, or
b) ASR substitutes the word with a phonetically similar alternative.

Phoneme-level, subword-level, and multilingual embedding-based methods mitigate—but do not eliminate—OOV detection difficulty [6], [26].

This is a major active research area.

9.3 Variability in Pronunciation, Accent, and Speaking Rate

Speech variation across:

1. speakers,
2. accents,
3. dialects,
4. speaking styles,
5. emotional states, and
6. spontaneous vs. read speech

makes STD significantly less reliable than text retrieval.

Even SOTA transformer models still struggle with:

a) heavily accented English
b) low-resource Indian languages
c) rapid code-switching
d) expressive emotional speech

This variability is one of the top obstacles cited in STD survey literature [12].

9.4 Background Noise, Reverberation & Channel Distortion

Real-world speech is contaminated by:

1. traffic noise
2. overlapping speech
3. broadcast distortions
4. microphone mismatch
5. reverberant indoor environments

Noise dramatically reduces:

1. ASR accuracy
2. posteriorgram reliability
3. embedding quality

PNCC, RASTA, CMVN help [22], but deep models still degrade under far-field or multi-speaker noise, especially in surveillance scenarios [18].

9.5 Multilingual and Code-Mixed Speech Complexity

STD systems assume:

a) consistent language usage
b) stable phoneme sets
c) predictable pronunciation rules

However, real-world speech often exhibits:

1. code-mixing (common in Indian languages)
2. language switching within a single utterance
3. foreign word insertion

Multilingual STD models such as Whisper [11] and Wav2Vec 2.0 XLSR [4] help, but true multilingual keyword search remains an open problem**.**

9.6 Lack of Temporal Localization in Text-Based STD

Acoustic STD returns start–end time boundaries. Text-based STD returns ranked documents without temporal spans.

This is a major limitation for:

1. forensics
2. surveillance review
3. media analysis
4. medical speech search

Unless alignment layers are added downstream, text-based STD cannot localize the keyword occurrence within the audio.

9.7 Limited Low-Resource Language Support

Most languages lack:

1. annotated speech data
2. pronunciation dictionaries
3. phoneme sets
4. labeled keyword examples

Zero-resource STD is a major research area [18], focusing on:

1. unsupervised phonetic unit discovery
2. speech embeddings
3. language-agnostic keyword spotting

Even state-of-the-art self-supervised models still perform sub-optimally on many African, Indian, and indigenous languages.

9.8 Domain Mismatch Between Training and Deployment

A model trained on:

1. broadcast news
2. academic lectures
3. scripted speech

fails when deployed on:

1. phone recordings
2. WhatsApp voice notes
3. street interviews
4. worn-body microphone feeds

All STD literature agrees: domain adaptation remains one of the hardest challenges [12], [14].

9.9 Poor Interpretability of Deep Models

Deep STD systems (especially transformer-based ones) act as black-boxes**,** complicating:

1. failure analysis
2. debugging
3. legal/forensic justification
4. bias detection

Interpretability is increasingly important due to ethical and regulatory requirements.

9.10 Computational Cost and Real-Time Scalability

Large-scale STD systems must handle:

1. thousands of hours of audio
2. millions of keyword queries
3. real-time or near-real-time latency constraints

Transformers and self-supervised models require:

1. high memory
2. GPU acceleration
3. significant inference time

This poses challenges for:

1. on-device STD
2. cloud cost optimization
3. edge intelligence applications

9.11 Benchmark Limitations (NIST, QUESST)

Benchmark datasets like:

1. NIST STD (2006–2010)
2. QUESST-2014

are:

[1] multilingual
[2] diverse

…but still limited in:

[3] noise variety
[4] spontaneous speech coverage
[5] conversational datasets

This restricts the generalizability of evaluation metrics such as ATWV and MAP across real-world scenarios [1], [29].

9.12 Lack of Unified Evaluation Standards for Modern STD

Classical metrics:

1. ATWV
2. Precision
3. Recall
4. MAP

struggle to capture:

[1] temporal accuracy
[2] semantic similarity
[3] cross-lingual detection
[4] near-misses vs. phonetically similar confusions

A unified evaluation framework for hybrid text–acoustic STD is still missing.

10. Open Research Problems in STD

Based on current literature [1], [4], [12], [18], [26], the major open research problems include:

10.1 Zero-Shot and Few-Shot Multilingual STD

Goal: detect keywords in languages with no training data.

Self-supervised models (Wav2Vec, HuBERT, Whisper) provide a starting point, but true zero-shot STD remains unsolved.

10.2 Stable OOV Keyword Detection

Accurate detection of:

a) unseen names
b) abbreviations
c) slang
d) new entities

is still largely unsolved for text-based STD.

10.3 Universal, Language-Agnostic Speech Units

A phonetic inventory that generalizes to all languages would revolutionize STD.

10.4 Domain-Robust and Noise-Robust STD

Transformers degrade under:

[1] far-field noise
[2] overlapping speakers
[3] emotional speech

Robustness remains a key research challenge.

10.5 Real-Time, Low-Power STD at the Edge

Running transformer-based STD on:

1. mobile phones
2. IoT devices
3. embedded hardware

is still heavily constrained due to memory/computation limitations.

10.6 Explainable and Interpretable STD Models

There is an urgent need for:

a) interpretable embeddings
b) transparent attention mechanisms
c) explainable keyword activation patterns

Especially for legal, medical, and forensic applications.

10.7 Unified Acoustic + Text-Based STD Systems

A model that jointly handles:

1. ASR transcripts
2. acoustic posteriorgrams
3. speech embeddings
4. multilingual units

does not yet exist. This is a major frontier**.**

10.8 Scalable STD for Massive Audio Repositories

Millions of hours of speech, such as call-center archives or surveillance audio, require:

1) efficient indexing techniques
2) approximate nearest neighbor search
3) hierarchical retrieval frameworks

Our Terrier-based pipeline is a strong step in this direction but remains text-dependent.

11. APPLICATIONS OF TEXT-BASED SPOKEN TERM DETECTION

Spoken Term Detection (STD) has transitioned from a research-focused technology to a critical component of modern speech-driven systems across surveillance, media analytics, healthcare, customer service, and digital information access. As speech becomes the dominant mode of interaction across devices and services, STD serves as a foundational technology enabling search, monitoring, indexing, detection, and retrieval from large-scale spoken archives.

This section presents a comprehensive application taxonomy, highlighting how STD is deployed across sectors and the role of text-based STD systems - such as the ASR → HappyScribe → Terrier → Hiemstra_LM framework—in real-world workflows.

11.1 Intelligence, Defense, and Security Surveillance

Surveillance remains the largest and earliest real-world application of STD. Intelligence agencies often receive massive volumes of:

1) intercepted communications,
2) call recordings,
3) radio transmissions,
4) field surveillance audio,
5) multilingual speech sources.

Manually analyzing such data is impossible. STD enables:

1) real-time detection of threat keywords
2) monitoring of suspicious conversations
3) screening for persons of interest
4) rapid filtering of intelligence databases

NIST specifically designed the STD evaluation tracks (2006–2010) to simulate intelligence use-cases, emphasizing ATWV-based error control [1], [29].

Posteriorgram-based STD approaches [2], [23] are particularly valuable in noisy or covert recordings.

11.2 Law Enforcement and Forensic Audio Analysis

STD supports:

1) Forensic investigation of criminal communication
2) Rapid retrieval of incriminating evidence from thousands of hours of police recordings
3) Keyword search in judicial proceedings, interrogation audio, and witness statements

Law enforcement often employs:

1) Phoneme-based STD for OOV names and rare event terms [6]
2) Noise-robust acoustic STD in low-quality field recordings [19]

Deep learning-based embedding systems (Wav2Vec 2.0, HuBERT) enhance forensic capabilities through improved robustness to distortion [4], [10].

11.3 Call Center Analytics and Customer Sentiment Monitoring

STD is a core technology in:

1) Business Process Outsourcing (BPO) analytics
2) Call monitoring for quality assurance
3) Compliance and fraud detection
4) Customer sentiment and escalation monitoring

STD enables:

1) Detection of "trigger terms" (refund, cancel, complaint)
2) Monitoring of agent compliance against regulatory scripts
3) Automatic identification of escalation points

Word-based STD (like our HappyScribe → Terrier setup) excels in this domain due to:

1) Large, structured speech corpora
2) High-quality ASR output
3) Domain-specific terminology

This domain is one of the largest commercial deployments of STD.

11.4 Broadcast Media Indexing and News Retrieval

Speech archives for news and broadcast media can span decades. Manual annotation is infeasible. STD enables:

1) Automatic keyword tagging of news broadcasts
2) Retrospective search for events, people, and places
3) Topic-based content retrieval
4) Multimedia archive management

The TREC Spoken Document Retrieval (SDR) tracks demonstrated the power of text-based STD for indexing large broadcast news collections [13].

IR-based approaches such as Terrier + Hiemstra_LM are particularly effective in this domain due to scalable ranking and MAP optimization [30].

11.5 Educational and Lecture Video Search

E-learning platforms (Coursera, NPTEL, tedX, YouTube EDU) generate thousands of hours of lecture recordings. STD enables:

1) Keyword-based lecture search
2) Automatic indexing of technical concepts
3) Rapid review of course content
4) Accessibility support for hearing-impaired learners

Transformer ASR models such as Whisper [11] dramatically improve lecture transcription quality, making text-based STD highly effective in academic settings.

11.6 Healthcare and Medical Dictation Retrieval

Medical professionals often record:

1) patient summaries,
2) clinical dictations,
3) diagnostic conversations,

4) case presentations.

STD helps in:

1) retrieving cases with specific symptoms or medications
2) compliance monitoring in clinical documentation
3) identifying safety-related keywords (e.g., “allergy,” “emergency”)

Due to the specialized vocabulary, OOV detection and domain adaptation are key research challenges [18].

11.7 Human–Computer Interaction and Voice Assistants

STD powers the “wake word” and command recognition systems behind:

1) Siri
2) Alexa
3) Google Assistant
4) Cortana
5) Smart appliances
6) Automotive voice control

Applications include:

1) Keyword-activated systems ("Hey Siri")
2) Command detection ("Call father", "Play music")
3) On-device keyword spotting for low latency

Deep CNNs, RNNs, and LSTMs dominate here [3], [8], [27], especially for embedded and streaming STD.

11.8 Multilingual Speech Access in Digital Libraries

National archives, parliamentary proceedings, historical interviews, and oral history collections rely on STD for:

1) Searching multilingual spoken documents
2) Accessing content across languages
3) Cross-lingual information retrieval

Multilingual self-supervised models (Wav2Vec 2.0 XLSR, Whisper) significantly expand STD’s usability for Indian and global low-resource languages [4], [11], [26].

11.9 Assistive Technologies for the Hearing Impaired

STD is used for:

1) Real-time keyword alerts
2) Visual highlighting of important speech segments
3) Smart captioning systems

These systems allow hearing-impaired individuals to detect:

1) alarms,
2) names being called,
3) emergency cues,
4) conversation shifts.

This aligns with global accessibility guidelines and represents a growing social-impact application.

11.10 Industrial and Manufacturing Monitoring

In high-risk industrial settings, STD is used to detect:

1) verbal safety commands
2) emergency signals
3) specific phrases indicating danger

Real-time STD ensures:

1) compliance with safety protocols
2) rapid response to critical events

Low-latency inference is essential, often requiring compact RNN or CNN models [27].

11.11 Social Media and Podcast Search Engines

Platforms such as YouTube, Spotify, and podcast search tools rely on STD for:

1) Content discovery
2) Topic classification
3) Keyword-based search within long audio streams

Podcast search is one of the most commercially successful deployments of transformer-based ASR → IR STD pipelines.

11.12 Why Text-Based STD (Terrier + Hiemstra_LM) Fits Many Applications

Our system is ideal for domains where:

1) Large-scale archives exist (broadcast, call centers)
2) ASR accuracy is reasonably high
3) Keyword queries are text-based
4) Fast document ranking is critical

Strengths:

1) High scalability
2) Efficient retrieval
3) Integration with IR tools
4) MAP optimization

Limitations:

1) No temporal localization
2) ASR dependency
3) OOV weakness

These constraints naturally fit commercial environments with high-quality speech capture.

12. FULL IMPLEMENTATION METHODOLOGY USING QUESST-2014, HAPPYSCRIBE ASR, TERRIER IR PLATFORM, AND HIEMSTRA_LM

This section describes the complete implementation framework proposed in this study for evaluating text-based Spoken Term Detection (STD). The methodology integrates:

1. QUESST-2014 multilingual STD dataset
2. Cloud-based ASR transcription using HappyScribe
3. Text normalization & document preparation
4. Terrier Information Retrieval indexing pipeline
5. Retrieval using the Hiemstra Language Model
6. Evaluation using Precision, Recall, and MAP

This implementation transforms **raw speech audio** into a **searchable textual corpus**, enabling scalable STD using established IR technologies. It also provides a reproducible baseline for future acoustic, hybrid, or multilingual STD research.

12.1 Overview of the Proposed STD Pipeline

The full processing chain is illustrated below:

QUESST-2014 Audio

↓

Speech Preprocessing

↓

HappyScribe ASR Transcription

↓

Text Normalization & Token Processing

↓

Terrier Inverted Index Construction

↓

Hiemstra_LM Retrieval for Keyword Queries

↓

Precision | Recall | MAP Evaluation

This hybrid design leverages deep learning–based ASR front-end with probabilistic information retrieval back-end, achieving scalability and robustness for spoken document retrieval tasks.

12.2 Dataset Description: QUESST-2014

QUESST-2014 (Query by Example Search on Speech Task, MediaEval 2014) is one of the most widely used multilingual STD benchmark datasets [26]. It contains:

1) Multilingual speech recordings (Slovak, Czech, Spanish, NNenglish, Albanian, Romanian)
2) Textual queries representing words or short phrases
3) Ground-truth relevance lists for MAP evaluation
4) Diverse recording conditions (studio, telephone, broadcast)

The dataset is designed to evaluate:

1) Keyword search
2) Cross-lingual and low-resource STD
3) Robustness to real-world noise conditions

This makes it well-suited for evaluating both classical STD approaches and IR-based systems such as the proposed implementation.

12.3 Speech Preprocessing

Before transcription, each audio file undergoes:

1. Amplitude normalization
2. Silence trimming via VAD
3. Noise-level equalization

Although HappyScribe internally performs preprocessing as part of its ASR pipeline, normalization is applied externally to ensure uniformity across QUESST recordings.

Preprocessing follows conventions established by NIST STD evaluations [1], [29].

12.4 Speech Transcription Using HappyScribe

Unlike traditional STD systems that require training ASR models, we employ **HappyScribe**, a commercial cloud-based ASR engine, to transcribe QUESST-2014 audio.

Rationale

1) Eliminates need for acoustic model training
2) Provides high accuracy using transformer-based ASR
3) Scales efficiently for hundreds of files
4) Supports multiple languages

HappyScribe's internal technology is proprietary but likely based on Log-Mel + Transformer architectures similar to Whisper [11].

Workflow

1. Upload each QUESST audio file to HappyScribe.
2. Export the resulting plain text transcription (UTF-8).
3. Associate each transcript with the original file identifier.

Output Quality

HappyScribe produces transcripts with:

1) Punctuation

2) Sentence segmentation
3) Word-level tokens

These are then normalized for IR indexing.

Limitations

All ASR errors propagate into text-based STD, as widely reported in ASR-driven STD literature [12], [18].

12.5 Text Normalization and Document Preparation

To prepare transcripts for IR indexing, the following steps are applied:

12.5.1 Tokenization

a) Split text into word tokens
b) Remove punctuation
c) Convert to lowercase

12.5.2 Stop-Word Removal

Common stop-words ("the", "is", "and") are removed based on multilingual stop-word lists.

12.5.3 Stemming / Lemmatization

Porter stemmer is used for English tokens; Snowball stemmers for other languages.

12.5.4 Document–ID Mapping

Each processed transcript is stored as:

<docno> FILE_ID
<text> processed transcription
</text>
</docno>

This format follows the TREC document standard, ensuring compatibility with Terrier [30].

12.6 Index Construction with the Terrier IR Platform

Terrier is one of the most widely used IR platforms for academic research, known for efficiency and extensibility [30].

Indexing Process

1. Specify the transcription directory in Terrier's collection specification file.
2. Run the indexer using:

bin/trec_terrier.sh –i

Terrier constructs:

a) Inverted index
b) Lexicon
c) Document length statistics
d) Term frequency tables

Retrieval Models Supported

Terrier supports:

1. TF-IDF
2. BM25

3. Hiemstra_LM
4. PL2
5. DFR models

Our system uses **Hiemstra_LM**, making it a probabilistic language-model-based STD approach.

12.7 Keyword Query Processing for STD

QUESST-2014 provides textual keyword queries**.**
For each keyword:

1. Remove diacritics (if needed)
2. Apply lowercasing
3. Apply same token processing as transcripts
4. Submit the token or token-sequence to Terrier as a textual query

Example query file entry:
<num> 001
<title> casa
</title>

This textual query is used to retrieve ranked speech documents.

12.8 Retrieval Using the Hiemstra Language Model

The Hiemstra Language Model (Hiemstra_LM) is a smoothed language model retrieval method suitable for keyword search in noisy transcripts [30].

Scoring Function

$$\mathbf{Score}(\mathbf{d},\mathbf{q}) = \sum_{t \in q} \log\big(\lambda P(t \mid d) + (1-\lambda) P(t \mid C)\big) \qquad (12.1)$$

Where:

P(t|d) = f(t, d) | |d| : term probability in the document
P(t|C): background model probability
λ: smoothing weight

Why Hiemstra_LM is Appropriate for STD?

1) Handles transcription noise better than TF-IDF
2) Smoothes rare keyword occurrences
3) Ranked retrieval aligns with MAP computation
4) Suitable for multilingual corpora

Terrier Configuration

In etc/trec/models.ini:
hiemstra.lambda=0.25
In trec.properties:
trec.model=Hiemstra_LM

This setup optimizes retrieval precision and reduces false alarms caused by noisy ASR.

12.9 Evaluation Metrics and Ground Truth Mapping

QUESST-2014 provides:

1) Document-level ground truth relevance lists
2) Query lists

3) Standard scoring scripts

Our system evaluates:

1) Precision
2) Recall
3) Mean Average Precision (MAP)

MAP is computed using Terrier's built-in TREC evaluation routines.

This evaluation protocol follows:

[1] TREC SDR standards [13]
[2] STD evaluation practices reported in NIST challenges [1]

12.10 Implementation Strengths

1) Scalable to large audio corpora
2) Minimal ASR system design required
3) Easy reproducibility
4) Compatible with multilingual data
5) Ideal for spoken document retrieval tasks
6) Uses well-established IR tools and probabilistic ranking models

12.11 Implementation Limitations

1) No temporal localization of detected keywords
2) OOV keywords cannot be detected
3) Pronunciation variation is not modeled explicitly
4) Completely dependent on ASR quality
5) Cannot perform acoustic matching or alignment

These are known weaknesses of ASR-driven STD pipelines [12], [18].

12.12 Reproducibility & Ethical Compliance

To ensure research transparency:

1) All HappyScribe transcripts can be archived
2) Terrier configuration files can be included
3) Query lists and relevance files are publicly available

Ethical considerations restrict uploading sensitive speech to cloud transcription services; however, QUESST-2014 contains no personal data.

13. FUTURE SCOPE FOR TEXT-BASED AND HYBRID SPOKEN TERM DETECTION

Spoken Term Detection (STD) continues to evolve rapidly due to advances in deep learning, self-supervised models, multilingual speech processing, and scalable information retrieval techniques. Although current STD systems—including the ASR-driven text retrieval framework presented in this study - provide strong performance for structured speech and clean audio, numerous research challenges remain open. This section outlines future research directions with scientific justification and references to emerging literature.

13.1 Integration of Acoustic and Text-Based STD for Hybrid Detection

Current text-based systems (HappyScribe → Terrier → Hiemstra_LM) rely entirely on ASR transcripts, which limits performance under:

1. ASR errors
2. OOV keywords
3. multilingual mismatch
4. noisy or conversational speech

A promising direction is hybrid STD, which integrates:

1. ASR transcripts
2. Phoneme posteriorgrams
3. Acoustic embeddings
4. Transformer-based speech encoders

Hybrid STD systems combine the precision and scalability of text-based retrieval with the robustness of acoustic matching [2], [23].

This direction may lead to:

a) Improved recall for difficult keywords
b) Cross-checking between multiple evidence streams
c) Unified probabilistic fusion models

13.2 Zero-Shot and Few-Shot Multilingual STD

Self-supervised models such as Wav2Vec 2.0 XLSR [4], HuBERT [10], and Whisper [11] enable speech representation learning across hundreds of languages without labeled data.

This opens transformative possibilities:

a) STD for languages with no ASR models
b) Cross-language keyword detection
c) Subword-level multilingual search
d) Universal phone-like unit discovery

Future systems may perform STD on:

1. Tribal languages
2. Low-resource Indian regional languages
3. Code-mixed speech without needing language-specific training.

This is one of the most important future research directions recognized in STD literature [18], [26].

13.3 Development of Universal, Language-Agnostic Speech Units

One grand research objective is to develop a universal phonetic inventory or phonological embedding space that works across languages.

Current limitations:

1. Phoneme sets differ between languages
2. OOV terms require language-specific G2P models
3. Cross-lingual pronunciation ambiguity remains unsolved
4.

Emerging research in unsupervised subword discovery [18] and multilingual transformers [11], [26] is laying the groundwork for universal units that would allow:

1. STD in unseen languages
2. Language-independent keyword modeling
3. High accuracy even without ASR

This direction could redefine STD entirely.

13.4 Noise-Robust and Domain-Robust STD Models

Real-world STD suffers under:

1. Far-field audio
2. Overlapping speakers
3. Reverberant rooms
4. Outdoor noise
5. Low-quality smartphone microphones

Future systems require:

1. Advanced speech enhancement
2. Domain adaptation
3. Robust feature learning (PNCC, RASTA, self-supervised)
4. Multi-domain pretraining

Self-supervised models provide a foundation for this, but more work is needed to handle extreme acoustic variability [4], [10], [22].

13.5 On-Device and Edge STD for Low-Latency Applications

Applications like:

a. Smart home devices
b. Wearables
c. Automotive systems
d. Industrial safety alarms

require **real-time STD** with:

1. Low latency
2. Minimal memory footprint
3. Low compute power

Future research may explore:

1. Model compression
2. Distillation for lightweight STD
3. Quantization of transformer encoders
4. Hybrid CPU–DSP STD pipelines

This represents a major engineering and modeling challenge.

13.6 Explainable and Interpretable STD Systems

STD models—especially transformers—operate as black boxes. This limits:

1. Debugging
2. Forensic use
3. Legal admissibility
4. Bias detection
5. Transparency in high-stakes applications

Future work should explore:

1. Attention visualization for keyword activation
2. Explainable relevance scoring
3. Transparent temporal localization models

Research in explainable AI (XAI) for speech is in early stages and presents a large opportunity.

13.7 Real-Time STD with Temporal Localization in IR-Based Pipelines

Current text-based STD (including your Terrier system) retrieves entire documents**,** not time segments**.** Next-generation IR-based STD frameworks must enable:

1. Time-stamped keyword localization
2. Passage-level indexing of transcripts
3. Hybrid IR + alignment mechanisms
4. Sentence-level retrieval

Combining Terrier with:
1. forced alignment tools,
2. Whisper timestamps,
3. VAD-based segmentation, could deliver IR systems with time-accurate STD.

13.8 Improving OOV Detection in Text-Based STD

To overcome OOV limitations, future systems should integrate:

1. Subword/BPE query expansion [4]
2. Phoneme-based fallback using G2P models [6]
3. Embedding-based similarity (cosine, dot-product) [10], [11]
4. Neural fuzzy matching

Such expansions would mitigate transcription failures and improve robustness for:
1. Named entities
2. Rare words
3. Dialect-specific terms
4. New vocabulary

13.9 Large-Scale, Distributed STD for Massive Audio Repositories

Institutions generate **petabytes of speech data**, including:
1. call-center archives
2. broadcast news
3. surveillance feeds
4. lecture collections

Future research must explore:
1. Distributed indexing
2. GPU-accelerated retrieval
3. Approximate nearest neighbor search
4. Hierarchical audio embedding architectures

Our **Terrier-based MAP retrieval** offers a baseline for scalable STD but further distributed enhancements are required for real-time big data environments.

13.10 Unified Evaluation Standards for Modern STD Systems

Current STD metrics (MAP, ATWV, Precision/Recall) do not adequately capture:
1. Temporal localization
2. Semantic similarity
3. Cross-lingual retrieval

4. Fuzzy matching behavior

Future work must develop:

1. Hybrid acoustic-text evaluation metrics
2. Localized MAP (L-MAP)
3. Cross-lingual STD benchmarks

The research community acknowledges this as a major gap [1], [12], [29].

14 Conclusion

This paper presented a comprehensive survey and a complete implementation framework for text-based Spoken Term Detection (STD), integrating classical signal processing principles, probabilistic information retrieval models, and modern deep learning insights. Beginning with a detailed examination of the STD research landscape—from DTW template matching and posteriorgram-based detection to CNN, LSTM, and transformer-driven approaches—the survey established the evolution of STD as a multidisciplinary problem spanning speech processing, machine learning, and information retrieval.

Traditional STD systems relied heavily on acoustic modeling and phoneme-level sequence matching [2], [23], but these approaches often struggled under real-world constraints such as noise, speaker variability, multilinguality, and limited resources [12], [18]. Modern deep learning models, especially self-supervised architectures such as Wav2Vec 2.0 [4], HuBERT [10], and Whisper [11], have reshaped STD by providing robust, multilingual, and generalizable speech representations. Nevertheless, speaker variability, ASR dependency, OOV limitations, and domain mismatch remain significant barriers to achieving consistent, scalable STD across diverse speech scenarios.

In this context, the proposed ASR-driven, text-based STD framework-built using HappyScribe transcription**,** Terrier IR indexing**, and** Hiemstra_LM retrieval - demonstrates a practical and efficient approach for large-scale spoken document retrieval. The methodology aligns with TREC-style evaluation workflows [13], leverages probabilistic ranking models well-suited for noisy ASR transcripts [30], and produces interpretable ranking outputs optimized through MAP, one of the most widely accepted measures of retrieval accuracy. By operating entirely at the text level, the pipeline enables scalability, reproducibility, and language flexibility without requiring complex acoustic model training.

The implementation using the QUESST-2014 dataset further validates the framework's applicability to multilingual STD benchmarking. However, as analyzed in earlier sections, the system inherits intrinsic limitations of ASR-based STD: lack of temporal localization, sensitivity to transcription errors, and inability to detect OOV keywords. These constraints mirror open research challenges documented extensively in the STD literature [6], [12], [18].

Despite these limitations, the framework provides a strong, foundational baseline for future research in hybrid STD systems that integrate acoustic embeddings, transformer features, and probabilistic retrieval. Opportunities for future work include developing hybrid acoustic-text STD, multilingual zero-shot keyword detection, cross-lingual embeddings, edge-optimized STD models, explainable attention-based retrieval systems, and enhanced benchmarking standards that reflect modern STD requirements.

In summary, this study contributes three major elements to STD research:

1. A comprehensive, up-to-date survey of STD techniques encompassing classical, deep, and self-supervised paradigms.
2. A scalable and reproducible STD implementation pipeline using widely accessible tools such as HappyScribe and Terrier.
3. A forward-looking analysis identifying critical research challenges and opportunities that will shape the next decade of STD innovation.

Together, these contributions offer both a conceptual roadmap and a practical methodology for researchers and practitioners aiming to develop the next generation of multilingual, robust, and scalable Spoken Term Detection systems.

15. References

[1] Fiscus, J. G., Ajot, J., Michel, M., and Garofolo, J. 2007. Overview of the NIST Spoken Term Detection Evaluation. *Proceedings of ICASSP*, pp. 399–402.

[2] Mangu, L., Cheng, C., and Padmanabhan, M. 2000. Finding Keywords in Continuous Speech Using Posterior Probability Estimation. *Proceedings of ICASSP*, pp. 129–132.

[3] Chen, S., Parada, C., and Heigold, G. 2014. Small-Footprint Keyword Spotting Using Deep Neural Networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(10): 1539–1543.

[4] Baevski, A., Zhou, H., Mohamed, A., and Auli, M. 2020. Wav2Vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *NeurIPS*, 33: 12449–12460.

[5] Hermansky, H. 1990. Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of the Acoustical Society of America*, 87(4): 1738–1752.

[6] Sun, M., Ma, W., and Fu, Y. 2016. Phoneme-Based Spoken Term Detection for Low-Resource Languages. *IEEE Signal Processing Letters*, 23(3): 374–378.

[7] Povey, D., Ghoshal, A., Boulianne, G., et al. 2011. The Kaldi Speech Recognition Toolkit. *IEEE ASRU Workshop*.

[8] Sainath, T. N., Mohamed, A., Kingsbury, B., and Ramabhadran, B. 2015. Convolutional Neural Networks for Large-Scale Speech Recognition. *Proceedings of ICASSP*, pp. 485–489.

[9] Graves, A., Mohamed, A., and Hinton, G. 2013. Speech Recognition with Deep Recurrent Neural Networks. *Proceedings of ICASSP*, pp. 6645–6649.

[10] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., et al. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *NeurIPS*, 34: 6555–6568.

[11] Radford, A., Kim, J. W., Xu, T., et al. 2022. Robust Speech Recognition via Large-Scale Weak Supervision (Whisper). *OpenAI Technical Report*.

[12] Jansen, M., and Watanabe, S. 2013. A Survey of Spoken Term Detection. *IEEE Signal Processing Magazine*, 30(2): 126–135.

[13] Anguera, X., Bullock, D., and Leiva, J. 2012. Spoken Document Retrieval for Speech Analytics. *IEEE Signal Processing Letters*, 19(12): 873–876.

[14] Nguyen, P., Church, K., and Stüker, S. 2018. Low-Resource Keyword Search Using Cross-Lingual Phonetic Embeddings. *INTERSPEECH*, pp. 2579–2583.

[15] Glass, J. 2003. A Segment-Based Approach to Speech Recognition. *Computer Speech & Language*, 17(1): 137–152.

[16] Zhang, Y., Suda, N., Lai, L., and Chandra, V. 2015. Deep Learning Based Keyword Spotting. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(11): 1870–1882.

[17] Veselý, K., Ghoshal, A., Burget, L., and Povey, D. 2013. Sequence-Discriminative Training of Deep Neural Networks. *Proceedings of ICASSP*, pp. 6840–6844.

[18] Thomas, S., and Jansen, A. 2017. Zero-Resource Speech Processing: A Review. *IEEE Signal Processing Magazine*, 34(5): 112–129.

[19] Raj, B., Singh, R., and Stern, R. 2010. Acoustic Keyword Spotting in Noisy Environments. *Proceedings of ICASSP*, pp. 4290–4293.

[20] Zweig, G., Nguyen, L., and Droppo, J. 2009. Speech Search in Low-Resource Languages. *INTERSPEECH*, pp. 440–443.

[21] Collobert, R., Weston, J., Bottou, L., et al. 2011. Natural Language Processing (Almost) From Scratch: A Unified Deep Learning Architecture. *ICML*, pp. 249–256.

[22] Cui, X., Zhang, S., Zhou, Z., and Zhao, X. 2016. Robust Acoustic Modeling Using Improved Feature Normalization. *IEEE Transactions on Audio, Speech, and Language Processing*, 24(2): 396–406.

[23] Karafiát, M., Grézl, F., Veselý, K., and Schwarz, P. 2011. Posterior-Based Approaches to Spoken Term Detection in Under-Resourced Languages. *Proceedings of ICASSP*, pp. 4852–4855.

[24] Livescu, K., Badaskar, S., and Bilmes, J. 2015. Learning Phonetic Representations Without Labeled Data. *Proceedings of ICASSP*, pp. 4380–4384.

[25] Trmal, J., Zhang, X., and Povey, D. 2017. The Kaldi OpenKWS Toolkit: Improving Low-Resource Keyword Search. *ASRU Workshop*, pp. 516–523.

[26] Rousseau, A., Železný, M., and Garner, P. 2014. Multilingual Spoken Term Detection Systems for QUESST. *INTERSPEECH*, pp. 2518–2522.

[27] Sak, H., Senior, A., and Beaufays, F. 2015. Long Short-Term Memory Recurrent Neural Networks for Keyword Spotting. *INTERSPEECH*, pp. 1478–1482.

[28] Bengio, Y., Courville, A., and Vincent, P. 2012. Deep Architectures for Speech Recognition: A Review. *IEEE Signal Processing Magazine*, 29(6): 82–97.

[29] NIST. 2007. Spoken Term Detection Evaluation Plan. *National Institute of Standards and Technology*, 1–50.

[30] University of Glasgow. 2022. Terrier Information Retrieval Platform. *Software Documentation*, School of Computing Science.