IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 November 2024, accepted 22 November 2024, date of publication 27 November 2024,
date of current version 9 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3506982

RESEARCH ARTICLE

Semantically Expanded Spoken Term Detection

ZHANIBEK KOZHIRBAYEV[1] **AND ZHANDOS YESSENBAYEV**[1,2]
[1]National Laboratory Astana, Nazarbayev University, 01000 Astana, Kazakhstan
[2]Computer Science Department, School of Computing and Creative Arts, Bina Nusantara University, Jakarta 10110, Indonesia

Corresponding author: Zhanibek Kozhirbayev (zhanibek.kozhirbayev@nu.edu.kz)

This work was supported by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan under Grant AP23489529.

ABSTRACT Spoken term detection (STD) is effectively implemented using fundamental techniques such as automatic speech recognition (ASR) and information retrieval. Through these methods, queried keywords can be identified in the decoded texts and indexed lattices produced by the ASR system. However, this approach relies heavily on the performance of the ASR; it may not produce the desired results when dealing with out-of-vocabulary (OOV) words that are not included in the ASR's lexicon. To address this limitation, we analyzed the semantic query expansion technique through extensive and reproducible experiments to assess its impact on the search quality for OOV words. We propose an approach to enhance existing spoken content retrieval methods by searching semantically expanded query sets and leveraging the advanced features of search engines. Our experiments, conducted on the Wall Street Journal (WSJ) datasets and top Google frequent queries, demonstrate that the proposed approach significantly improves retrieval accuracy over the traditional word-based STD method for in-vocabulary (IV) terms. Specifically, the Actual Term Weighted Value (ATWV) score improved from 0 to 0.5776 for the trigram query category. Additionally, our approach outperforms the proxy-based method for OOV words. While the proxy-based technique fails to retrieve results for both bigrams and trigrams, the semantic-based approach achieves ATWV scores of 0.7143 and 0.8846 for bigrams and trigrams, respectively. Furthermore, substantial gains are observed when combining semantic-based query expansion with a full-text search engine, improving the performance of the word-based STD system by approximately 3 to 4 times on the bigram and trigram query categories.

INDEX TERMS Keyword spotting, query expansion, semantic retrieval, spoken content retrieval, spoken term detection.

I. INTRODUCTION

Data science has become a cornerstone of the information technology industry, with recorded materials serving as a vital resource in this field. Advances in data processing techniques—particularly in indexing, retrieval, search, and browsing of media content—have not only streamlined various tasks but also fueled the rapid expansion of internet-based content and services. Despite the fact that currently multimedia content is increasing gradually with broadcast news from channels, radios, social media and so on, it is still generally processed mainly on the basis of manually created textual metadata and description. Given that text and speech are alternative forms of expressing human language, it is imperative that both modalities receive equal attention in processing. As the volume of audio data grows, so does the need for robust and efficient information retrieval systems tailored to audio data storage. Spoken Term Detection (STD) is designed to meet this demand by efficiently and accurately identifying specific terms—whether single words or phrases—within extensive and varied audio repositories. Unlike Spoken Document Retrieval (SDR), which retrieves entire documents containing the query terms, STD focuses on detecting each instance of a queried keyword, including its precise start and end times within the spoken content. The general framework of the STD system can be described as follows (Fig. 1). The system processes two key inputs: raw audio data and a list of search keywords. It operates in two main phases: indexing and searching. In the indexing phase, audio data is converted into word lattices using an

The associate editor coordinating the review of this manuscript and approving it for publication was Ganesh Naik.

automatic speech recognition (ASR) subsystem. These word lattices are then indexed independently, allowing the system to process audio data without prior knowledge of the search terms. During the searching phase, the indexed word lattices are matched against the search queries using a word-based STD module. The system leverages the pre-constructed index to retrieve relevant terms, assigning a confidence score to each detection to indicate its accuracy. This phase can be repeated for multiple queries, underscoring the importance of efficient and scalable search processes.

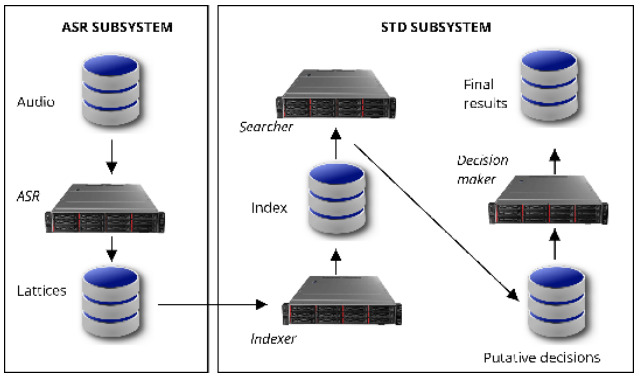


FIGURE 1. Spoken term detection system architecture.

The originality of this work lies in its novel and easily integrable approach, combining semantic-based query expansion with advanced full-text search engine features, such as lemmatization and complex search syntax, for spoken term detection. This enables the system to generate query variations by considering synonyms, improving the retrieval of semantically similar terms and effectively addressing OOV challenges. Additionally, the research provides a detailed analysis of the impact of different query sizes (unigrams, bigrams, trigrams) on retrieval performance. The method is simple to implement, deploy, and integrate into existing complex search engine systems, and its robustness is validated through fully reproducible experiments, showing superiority over traditional word-based methods.

We conducted detailed experiments assessing the impact of semantic query expansion compared to word-based methods and how our system addresses the out-of-vocabulary issue. Our contributions to the research are as follows:

- We performed thorough and fully reproducible experiments evaluating various search techniques, including semantic query expansion, to ensure reliability in our findings.
- We analyzed the effect of query size (unigrams, bigrams, and trigrams) on search results, providing insights into how different n-gram approaches influence retrieval effectiveness.
- We experimentally demonstrated that our method outperforms the proxy-based approach in effectively handling OOV words, highlighting the robustness of our approach in real-world applications.
- We explored the integration of semantic knowledge into our query expansion process, enhancing the relevance of search results.

This paper is organized as follows: Section II provides an overview of various techniques for spoken term detection. Section III details the four main components of our system: speech recognition, word-based STD, semantic-based query expansion, and full-text search engine integration. The evaluation corpora and methodology are described in Section IV. Section V presents the experimental setup, along with the results obtained from our proposed spoken term detection system. In addition, it shows the processing resource measurements, describing how a single request is processed using the proposed approach. Finally, Section V-C summarizes the findings and conclusions drawn from the experiments and outlines potential directions for future research.

II. RELATED WORK

Considerable research has been conducted on the STD task. Much of this work has focused on improving the accuracy of results rather than addressing the challenge of OOV terms. Previous studies have shown that traditional methods for detecting OOV terms, such as proxy keywords [1] or approaches based on phones, syllables, and morphemes [2], [3], [4], often fall short in effectively solving this issue. Consequently, there is still significant potential to explore new techniques for handling keywords that are absent from the lexicon. In many cases, the exact query terms may not be present in the spoken content, yet semantically similar terms might exist that fulfill the search request. For example, if the query contains the phrase *"obstinate person,"* the system could retrieve not only exact matches but also semantically related terms like *"stubborn person."* While various methods for semantic extraction have proven effective in such scenarios, relatively few approaches have been specifically tailored or extensively explored in the context of audio data.

Speech processing faces significant challenges in large vocabulary continuous speech recognition (LVCSR) as vocabulary sizes increase. The extensive vocabularies, frequent occurrence of OOV words, and sparse data available for language models (LM) are major limitations of word-based lexical approaches. Extensive research has been undertaken to address these challenges, particularly those related to the OOV issue. This research draws from various fields, including speech recognition, spoken term detection, and language modeling. In this section, we present a review of some related work, focusing specifically on methods for handling OOV words.

Subword-based methods have shown significant effectiveness in handling OOV terms. As explored in several studies [2], [3], [4], subword units such as word fragments, acoustic words, multigrams, syllables, and graphemes are considered among the most effective techniques for OOV detection. In various works utilizing subword units [1], [3],

[4], the approach typically involves searching for the exact sequence of these units. Other studies [5], [6] propose using a confusion model to create ''proxy keywords,'' enabling the search to include words that sound similar. The search process for proxy keywords is similar to that of in-vocabulary (IV) words in word-level lattices, with the added step of generating proxies [7]. Another approach involves using a confusion network (CN) [8], where a word index based on confusion networks is argued to be sufficient for high-performance open vocabulary term retrieval. A hybrid technique [9] has also been explored, combining word-based and subword-based methods to leverage the strengths of both approaches. Additionally, hierarchical n-gram LMs that incorporate both words and characters have been investigated to improve OOV word detection [10]. A common technique for addressing OOV terms involves the proactive expansion of the LVCSR dictionary. Specifically, additional pronunciations for a large number of words are pre-structured within the LVCSR dictionary to generate lattices [11]. A comprehensive overview of spoken content retrieval concepts is provided in [12], which outlines five primary directions, including semantic retrieval. The authors argue that fundamental models, such as the vector space model and language modeling retrieval approaches, can be effectively applied to spoken content as well.

Several studies have employed machine learning and neural network techniques to enhance the accuracy of STD. For instance, [13] suggests that acoustic distances between subwords and Hidden Markov Model (HMM) states, where the posterior probabilities are derived from a deep neural network, can be leveraged to improve STD performance. A method utilizing single-pass decoding with a bidirectional long short-term memory (BLSTM) network was introduced in [14]. Additionally, the study in [15] explores the use of recurrent neural networks (RNNs) during training, examining the effects of different loss functions, the volume of unsupervised data, and meta-parameters. Reference [16] presents an approach based on a classifier trained using machine learning techniques combined with the dynamic time warping (DTW) method, which can be applied to term detection in zero-resource languages. Another study, [17], integrates neural networks into acoustic modeling and evaluates the entire pipeline with respect to keyword search (KWS) performance.

The OOV issue can also be addressed by leveraging term semantic relationships. Some approaches have achieved significant performance improvements using co-occurrence data and hand-crafted thesauri, while others have utilized resources like Wikipedia, Wiktionary, or WordNet to generate queries that are semantically related to the user's key query. For instance, [18] argues that the results of language modeling approaches can be enhanced by applying query expansion based on co-occurrences. The methodology of enriching the corpus by expanding the original collection with WordNet terms was proposed in [19]. A novel approach for collecting data to retrieve spoken utterances that are semantically relevant to a given text query was introduced in [20]. Several studies [21], [22] propose semantic models for concept representation, while [23] explores query expansion methods such as pseudo-relevance feedback and word embeddings, demonstrating that the combination of these techniques yields significantly improved performance. There are also ontology-based techniques which have been shown very effective with semantic-aware text-based search. This work [24] introduces SemIndex, a framework that integrates a semantic network with an inverted index to improve the matching of user queries with data terms. The authors of this paper [25] present an ontology-based semantic search method for open government datasets, utilizing hybrid indexing and natural language processing to enhance search efficiency. A deep learning-based system that employs a Triplet-BERT model for semantic search tasks, emphasizing its value in concept normalization and ontology matching was presented in this research [26].

We also analyzed commercial systems which provide spoken term detection such as Panopto [27], MAVIS [28], Remeeting [29], etc. The research paper [30] states that MAVIS, which is an audio and video indexing service running in Windows Azure, utilizes metadata of a media file in order to generate keywords that are sent to a search engine to get related content. Remeeting claims that its search indexing uses Elasticsearch features and semantic and phonetic query expansion to improve the search performance [31], [32].

Although their approach may appear similar to ours, we were unable to find detailed information on experiments that assess the impact of semantic query expansion compared to word-based methods, or on how their system addresses the OOV issue.

III. METHODOLOGY

Our spoken term detection system is comprised of four main components: ASR, word-based STD, semantic-based query expansion, and a full-text search engine. Initially, speech data is converted into word lattices and decoded texts. These outputs are then indexed separately. The query set is prepared by applying semantic-based query expansion. During operation, the word lattices and query set are processed by the word-based STD unit, while the decoded texts and query set are sent to the full-text search engine. Subsequently, all matched occurrences of the query terms found in the resulting documents are processed by the STD system to determine their start and end times (see Fig. 2). Each component is described in detail below.

A. SPEECH RECOGNITION

The ASR unit utilizes the Kaldi toolkit [33] to generate word lattices from the raw audio data. Our LVCSR system is built by training deep neural networks (DNN) over of Feature space Maximum Likelihood Linear Regression (fMLLR) features. The training is conducted in 3 stages. A stack of

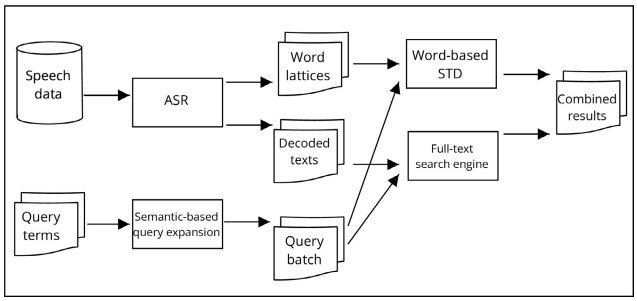


FIGURE 2. Units of the proposed spoken term detection system.

restricted Boltzmann machines (RBMs) was trained at the first unsupervised stage. This gives a good starting point for the next stage, which is called the frame cross-entropy training. The purpose of this stage is to classify frames into the appropriate probability density functions (pdfs). The purpose of the final stage, sequence-training optimizing state-level minimum Bayes risk (sMBR), is to accentuate state-sequences with better frame accuracy with respect to the reference alignment. Stochastic gradient descent with per-utterance updates and acoustic scale of 0.1 were applied at this stage. The lattices are re-generated after first epoch in order to get faster convergence. More detailed information can be found in [33].

We also built Subspace Gaussian Mixture Model (SGMM) retrained using (boosted) Maximum Mutual Information (MMI) as a sequence level objective. All these systems decode and generate word lattices for the same search collection, which will then be sent to the STD unit for indexing and searching. We select the best model with the lowest word error rate (WER) performance. The common metric of the performance of experimental speech recognition models is WER, which is computed as the ratio of erroneously recognized words to the total number of words.

The WSJ (Wall Street Journal) corpus [34] is used to train the acoustic model described above. This corpus consists of text and audio data derived from sentences read from the Wall Street Journal, recorded under consistent environmental conditions. It features a large vocabulary and contains approximately 80 hours of training data. The test materials are divided into utterances with and without verbal punctuation. Each WAV file is accompanied by a TXT file containing detailed transcriptions and a time-aligned annotation file with gender-specific labels. Specifically, the discs include complete orthographic transcriptions of the speech data, as well as bigram language models for the Wall Street Journal text data from which the prompting text was sourced. The dictionary and language model (LM) for the ASR subsystem are based on this corpus. The test materials on these discs support 5,000-word and 20,000-word WSJ vocabulary continuous speech recognition (CSR) tests, as well as tests involving spontaneous dictation. Experimental results are summarized in Table 1.

TABLE 1. Minimum value of the WER.

Experiment Set	dev93	eval92
DNN on top of fMLLR features	5.96	3.42
SGMM + MMI	6.62	3.88

The LVCSR system is built by training DNN over fMLLR features show 5.96 and 3.42 as a WER score on dev93 and eval92 sets of the WSJ corpus. Models trained using SGMM and MMI show slightly worse WER results than the deep neural network model.

B. SPOKEN TERM DETECTION

Word lattices produced by the LVCSR system are processed using the lattice indexing approach outlined in [35]. Our STD subsystem, based on the Kaldi term detector, searches for keyword terms within the indexed lattices. Initially, word lattices from all utterances in the speech data, derived from individual weighted finite state transducers (WFSTs), are converted into a single generalized factor transducer pattern. This pattern consolidates the start and end times, along with the lattice posterior probabilities of each word token, into a three-tuple cost format as described in [35]. The factor transducer then constructs an inverted index of all word sequences contained in the lattices. With the search collection prepared, an ecf (evaluation control file) file is generated to store metadata, including the source signal duration, language, and dataset version. For example:

```
<ecf source_signal_duration="284400.825" language=
"english" version="WSJ dataset">
  <excerpt audio_filename="48uc0202" channel="1"
  tbeg="0.000" dur="12.46" source_type="splitcts"/>
  <excerpt audio_filename="49pc0402" channel="1"
  tbeg="0.000" dur="7.45" source_type="splitcts"/>
  ...
</ecf>
```

In this example, each <excerpt> tag specifies an audio segment with attributes like audio_filename, channel, tbeg (start time), dur (duration), and source_type. This information helps track the details of each audio file used in the search collection.

Following this, an rttm (rich transcription time marked) file is generated for scoring purposes. This rttm file is produced by force-aligning the search collection with a trained model, ensuring precise alignment for each lexeme. For instance:

```
LEXEME 46vc0402 1 0.000 0.230 center lex 46vc0402
<NA>
...
LEXEME 4anc0403 1 0.230 0.100 letter lex 4anc0403 <NA>
```

Here, each line in the rttm file represents a lexeme, providing information such as the lexeme label (LEXEME), audio filename, channel, start time, duration, and additional attributes. These attributes allow for detailed scoring of the

alignment process, ensuring that each term in the search collection is accurately positioned and time-stamped.

For a given search term, an ordinary finite state machine is created and integrated with the factor transducer to locate all occurrences of the term in the audio data. The posterior probabilities for the lattice with respect to all words in the search term are accumulated, assigning a confidence score to each detection. The decision-making process filters out detections with confidence scores below a predetermined threshold. For example, For the query ''consumer product,'' the system first generates a keyword list (kwlist) file and then constructs a finite state machine using a factor transducer. The system identifies all paths within the transducer that match the query ''consumer product,'' outputting the utterance ID, start time, end time, and posterior probability for each occurrence. All occurrences are then sorted by their posterior probabilities, ensuring that the most relevant results appear first. The results are formatted as follows:

```
<detected_kwlist kwid=''WSJ-0069'' search_time=''1''
oov_count=''0''>
   <kw file=''48uc0202'' channel=''1'' tbeg=''4.36'' dur=
   ''1.15'' score=''0.989383'' decision=''YES''/>
   <kw file=''49pc0402'' channel=''1'' tbeg=''2.78'' dur=
   ''0.84'' score=''0.97898'' decision=''YES''/>
   <kw file=''47wc0202'' channel=''1'' tbeg=''3.30'' dur=
   ''0.88'' score=''0.968783'' decision=''YES''/>
</detected_kwlist>
```

The primary challenge in the STD system is the detection of OOV words. If a term is not included in the ASR dictionary, it will be absent from the indexed lattices, even if it is present in the audio recording. Several approaches address this issue, including methods based on subwords, phonemes, and proxy keywords, among others.

1) WORD AND PHONEME BASED METHODS

The algorithms for developing STD systems based on words and phonemes are similar, except for the lattice structures used. Lattices are generated by the speech recognition system and indexed using the mechanism described in [35]. Queries can consist of single or multiple words. In the word-based approach, speech data is pre-indexed in terms of word units, allowing for the search of arbitrary terms without referring back to the original audio. In contrast, the phoneme-based approach involves searching for keywords within indexed phoneme lattices. Results from both approaches are ranked according to posterior probabilities [36].

2) PROXY KEYWORDS METHOD

Another technique [1] for searching for an OOV term can be created based on proxy words. This method focuses on finding words which sounds acoustically close to the given OOV word from the main lexicon dictionary and retrieve these proxy words instead of providing zero results. The benefit is that there is not necessity to create another subword system.

Let K stands for a finite-state acceptor for an OOV keyword, and L_2 for a finite state transducer, that keeps the pronunciation of K. More precisely, If K is not in main dictionary, this lexicon can be generated by using G2P tools. Let E' be an edit-distance transducer which keeps the phone confusions gathered from training set. Let L_1 represents the original lexicon of the ASR system. The way to build a proxy keyword K' may be formulated as,

$$K' = Project(ShortestPath(Prune(Prune(K \circ L_2 \circ E') \circ L_1^{-1}))). \quad (1)$$

In this equation, $(K \circ L_2 \circ E')$ represents the composition of the OOV acceptor with its pronunciation and the edit-distance transducer. This composition is then matched against the inverse of the original lexicon L_1^{-1}. The application of the *Prune* and *ShortestPath* operations ensures that the best candidate words are selected, and the *Project* operation extracts the desired proxy keyword.

Kaldi provides proxy search capabilities for OOV keywords within its STD search framework. Proxy search relies on a precomputed confusion matrix, which is specific to the training corpus, and a trained grapheme-to-phoneme (G2P) model. The G2P model is used to generate pronunciations for OOV keywords, allowing the system to identify phonetically similar matches.

C. SEMANTIC-BASED QUERY EXPANSION

In this section, we demonstrate how to integrate our proposed method, semantic-based query expansion, into the Kaldi-based keyword spotting approach. As discussed earlier, the word-based keyword spotting approach using Kaldi involves searching for keyword terms within lattices generated by the ASR subsystem. Compared to other retrieval models, such as subword-based and proxy-based approaches, this method has the advantage of detecting all spoken terms that are present in the ASR system's lexicon. However, the performance of the word-based method is limited by its reliance on exact term matching between queries and lattices, which does not accommodate semantically related terms. The primary challenge is that the ASR-generated word lattices may not always match the exact terms used in a user's query, leading to less effective spoken term detection. For instance, given the query phrase *"car transportation,"* apparently, word lattices of one audio record with the term *"vehicle transportation"* should have a higher score because *"car"* is semantically more related to *"vehicle."* However, Kaldi-based keyword spotting approaches do not support this type of semantic matching and would consider these documents as not matching the query. Although techniques such as query expansion and pseudo-relevance feedback offer potential for semantic term matching, query expansion based solely on semantic relationships has so far seen limited success, likely

due to the challenge of assigning relevant weights to new terms.

We propose to generate semantically related alternatives of given queries to increase the performance as well as deal with OOV terms, because word lattices may not contain exact queried terms or queried terms may not appear in the lexicon dictionary of the ASR system. With the intent to generate queries that are semantically related to key query given by user, sources of semantic information can be exploited such as Wikipedia, Wiktionary or WordNet. We use WordNet [37], an online lexical database for the English language, since synonym sets in WordNet (synsets) allow generating lexical paraphrases without writing exhausting codes in individual systems. For instance, for the given query *"modern technique,"* we can generate the paraphrase *"modern method"* simply by replacing the word *"technique"* in the sentence with its synonym method listed in WordNet synset. Whereas, such alternatives are not always semantically correct. For instance, *"tally"* and *"rack up"* are appeared in the synonyms list to the word *"score."* Despite the fact that the sentences like *"he scored 2 goals"* occur in newspaper sport reports more often, sentences like *"he tallied 2 goals"* or *"he racked up 2 goals"* almost never happen. In order to effectively apply WordNet for generating semantic alternatives of given queries, we need to create methods that can appropriately determine exchangeability of synonyms in a particular context.

For each given query, the system first generates all possible variations regarding the synonyms of each term in the query. Since this set of generated queries was too big and most of them are not semantically correct, we selected a small subset of these queries by applying the most frequent bigram dictionary from the Corpus of Contemporary American English [38].

We briefly summarize the high-level steps involved in the proposed method:

1) Find all possible synonyms of each term in given query using WordNet and construct set of variations.
2) Check if the bigram of each variation appear in the most frequent bigram dictionary. If it meets the condition, this variation will remain in the set; otherwise it will be removed from the set.
3) Apply batch mode to retrieve the results for the set of semantically related queries.

If the given query phrase consists of one word, we skip the second step.

D. FULL TEXT SEARCH ENGINE

Given a set of word lattices and an input query that is represented in a text form, the challenge of spoken term detection is retrieving the most relevant word lattices with the exact position in time measure of queried terms. Since the ASR unit generates decoded text as an output, we propose to integrate a full-text search engine with the word-based STD. As a result, our algorithm can return documents that have words in common with the input query. To do this, we used the tool SphinxSearch [39]. In our case, we used lemmatization, which replaces all found words with their normalized form, thus allowing, for example, on the request of the ''go'' to find documents containing the words ''goes'', ''going'' or ''went''.

Standard STD task finds contiguous interval with starting and ending times matching the query terms as a whole, whereas conventional search engine can find documents which contain query terms in any position. Clearly in some cases we need to retrieve audio file where query terms may occur apart from each other, but preserving their relative positions. For example, for the query ''black car'' we might find audio file with transcription as ''my black lovely car.'' Thus mixing the STD with the conventional search engine allows us to handle these cases. For such a use case, we can estimate the detected location of the matching in two ways:

- the start time and end time of each word: ''black'' and ''car'';
- the start time of the first word and end time of the last one, since we preserve positions of query terms: ''black lovely car''.

IV. EXPERIMENTAL SETUP

This section presents preprocessing stages such as test set preparation for STD and processing resources. Furthermore, it describes the STD evaluation methodology.

A. EVALUATION CORPORA AND TERMS

The evaluation of the STD system was carried out using the WSJ corpus, which was also used for the ASR system. Test search queries were sourced from the top Google frequent queries [40], in order to reflect real queries generated by human users. These queries were categorized into three types: unigrams, bigrams, and trigrams. We selected 800 queries for each category, comprising 270 trigrams, 270 bigrams, and 260 unigrams. Among these, 40 unigram queries, and 30 queries each for bigrams and trigrams, included at least one OOV word in any position. Our proposed method was applied to generate semantically related alternatives for both in-vocabulary (IV) and OOV queries (see Fig. 3 and 4). For IV queries, we utilized the word-based method, while for OOV queries, we used the proxy-based method to generate fuzzy alternatives. Additionally, the semantic-based query expansion approach was applied to both IV and OOV queries.

B. STD EVALUATION METHODOLOGY

The STD task involves detecting each queried keyword and providing its start and end times in the spoken content. The efficiency of STD systems is evaluated based on two types of errors: false alarms (FA) and missed detections (Miss). The National Institute of Standards and Technology (NIST) has established a pilot evaluation method for STD [41]. This method assesses system performance using two key approaches: Detection Error Tradeoff (DET) curves and Term-Weighted Value (TWV). DET curves provide a comprehensive view of system performance across varying

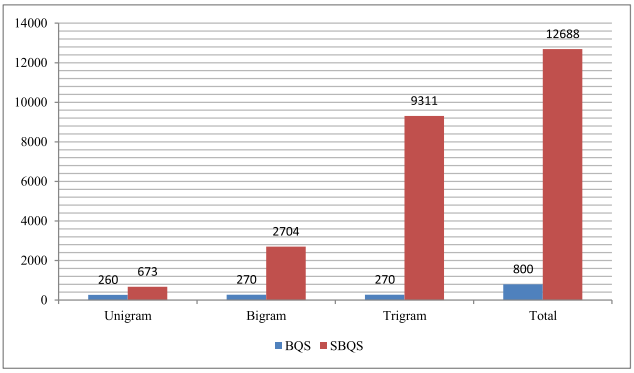


FIGURE 3. Size of IV query sets (BQS: basic query sets; SBQS: semantic-based query sets).

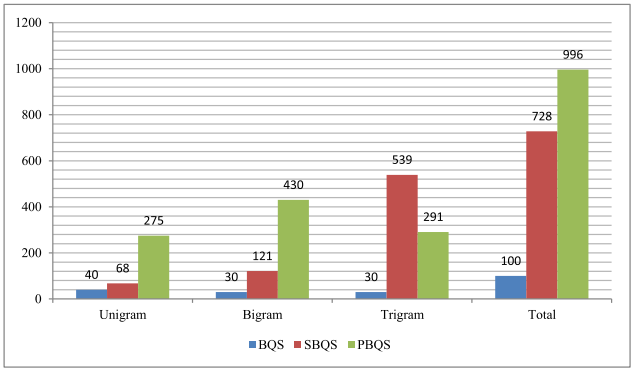


FIGURE 4. Size of OOV query sets (BQS: basic query sets; SBQS: semantic-based query sets; PBQS: proxy-based query sets).

recall and precision levels, while TWV offers a metric for system optimization by weighing the importance of detected terms.

1) DETECTION ERROR TRADEOFF CURVES

A detection error tradeoff (DET) curve is a graphical plot of error rates for binary classification systems, depicting the false rejection rate vs. false acceptance rate. It marks threshold, θ, detection functions called miss (PMiss) and false alarm probabilities (PFA) on a chart. This threshold is introduced to the detection scores (DS) that are determined individually for every queried term and averaged to produce a DET plot. A single term's PMiss and PFA can be computed using these formulas [41]:

$$P_{Miss}(term, \theta) = 1 - \frac{N_{correct}(term, \theta)}{N_{true}(term)}, \tag{2}$$

$$P_{FA}(term, \theta) = \frac{N_{spurious}(term, \theta)}{N_{NT}(term)}, \tag{3}$$

where:

- $N_{correct}(term, \theta)$ is the number of correct detections of term with $DS \geq \theta$;
- $N_{spurious}(term, \theta)$ is the number of incorrect detections of term with a $DS \geq \theta$;
- $N_{true}(term)$ is the true number of occurrences of term in the corpus;
- $N_{NT}(term)$ is the number of opportunities for incorrect detection of term in the corpus.

2) TERM WEIGHTED VALUE

A perfect system always responds correctly to a stimulus; however an omitted response or a misleading response reduces the value of a system to a user. Therefore, Term-Weighted Value (TWV) represents one minus the average value lost by the system per term. The value lost by the system can be defined as a weighted linear combination of miss and false alarm probabilities as mentioned above. The weight, β, considers both the prior probability of a term and the relative weights for each error type. The ratio equal to 1.0 for TWV represents the value of system as an ideal without misses and false alarms.

Term-Weighted Value (TWV) can be calculated using the following formula [41]:

$$TWV(\theta) = 1 - \underset{term}{average}\{P_{Miss}(term, \theta) + \beta \cdot P_{FA}(term, \theta)\}, \tag{4}$$

where:

$$\beta = \frac{C}{V} \cdot (Pr_{term}^{-1} - 1). \tag{5}$$

θ is the detection threshold.

For the current evaluation:

- the cost/value ratio, $\frac{C}{V}$, is 0.1,
- the prior probability of a term, Pr_{term}, is 10^{-4},
- using the given values, the weight, β, comes out to be approximately 999.9, indicating that a false alarm is much less costly than a miss, given the term's low prior probability.

3) ACTUAL TERM WEIGHTED VALUE

Actual Term Weighted Value (ATWV) is considered as a primary metric for the evaluation. It combines the hit rate and false alarm rate of each *kw* into a single measure and then averages across all the terms. NIST defined ATWV as the value of TWV at the system-selected detection threshold, $\hat{\theta}$, i.e., $ATWV = TWV(\hat{\theta})$.

4) MAXIMUM TERM WEIGHTED VALUE

Maximum Term Weighted Value (MTWV) provides an upper bound TWV in the case that it has an ideal global threshold. A search will be made for the maximum TWV among all possible *theta* values while analysing the DET curve of the system. The difference between ATWV and MTWV is a measure of how well the hard decision threshold was installed.

5) OPTIMAL TERM WEIGHTED VALUE

Optimal Term Weighted Value (OTWV) provides an upper bound on system performance in the case that it has an ideal threshold value depending on *KW*. The detection threshold $\hat{\theta}_{kw}$ which makes great as possible the keyword-specific *TMV*

or keyword's $P_{Miss}(kw, \theta)$ and $P_{FA}(kw, \theta)$ can be selected for each kw:

$$\hat{\theta}_{kw} = \min_{\theta}(1 - [P_{Miss}(kw, \theta) + \beta \cdot P_{FA}(kw, \theta)]) \quad (6)$$

Let kw represent a keyword, and let K denote the total number of keywords. Then the OTWV can be defined with this formula:

$$OTWV = [\sum_{kw=1}^{K} \frac{P_{Miss}(kw, \hat{\theta}_{kw})}{K} + \beta \sum_{kw=1}^{K} \frac{P_{FA}(kw, \hat{\theta}_{kw})}{K}] \quad (7)$$

V. EXPERIMENTS AND RESULTS

This section presents the experimental results. The first experiment evaluates the performance of the word-based and proxy-based STD systems. The second experiment assesses the overall impact of incorporating full-text search compared to the word-based STD system.

A. EXPERIMENT I: SPOKEN TERM DETECTION

In this experiment, we evaluate the performance of the STD system based on Kaldi using both IV and OOV queries. The results are presented in Tables 2 and 3, which show the performance of the STD system in terms of the ATWV score.

TABLE 2. STD performance for different IV keyword categories.

Metric	Word-based method			Semantic-based method		
	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram
ATWV	0.7956	0.8248	0.0	0.7635	0.8696	0.5776

TABLE 3. STD performance for different OOV keyword categories.

Metric	Proxy-based method			Semantic-based method		
	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram
ATWV	0.9548	0.0	0.0	0. 7628	0.8508	0.5937

The semantic-based query expansion method yields significant improvements over the basic query set for IV terms (Table 2). Notably, for the trigram query set, there is a marked improvement with an ATWV score increase from 0 to 0.5776. Substantial gains are also observed in the bigram query set, with the ATWV score rising from 0.8248 to 0.8696. However, there is a slight decrease in the ATWV score for unigram queries. Table 3 presents the ATWV score for OOV queries. The search for the expanded queries on the basis of proxy does not retrieve any results for both bigrams and trigram, whilst the semantic-based approach shows 0.7143 and 0.8846 ATWV scores for bigrams and trigrams, respectively. On the other hand, the ATWV score for unigram semantic-based query set decreased compared to proxy-based query set. The reason for that seems to be followings. The proxy-based approach generates similar terms that are in vocabulary resulting in high level of hits, while the semantic-based approach generates the terms that may not be in vocabulary resulting in high misses. From the experiments we also note that it is hardly true that the generated word lattices from the ASR subsystem always have exactly the same terms as a user would use in a query. This kind of constraint makes the spoken term detection performance of existing method non-effective.

The DET plots of each query category can be seen in Fig. 5 - 8. The graphs show TWV, tradeoff between miss (PMiss) and false alarm (PFA) probabilities, maximum TWV and threshold score for each query category. This threshold is determined empirically for every queried term and averaged to produce a DET plot. The decision of the occurrences having a score less than threshold is set to false; false occurrences obtained by the system are not taken into account as retrieval. MTWV is the TWV at the point on the DET curve where a value of threshold θ produces the maximum TWV. The maximum possible value for TWV is 1.0, which corresponds to "perfect" system result. We append 'U_' to the unigram, 'B_' to the bigram and 'T_' to the trigram query sets, e.g. U_TWV. The MTWV scores by the semantic-based method overcome the performance shown by the word-based and proxy-based methods for IV and OOV queries, respectively, in bigram and trigram categories. Regarding OOV terms, the proxy-based method for unigram category shows 0.9548 on MTWV, which is higher than semantic-based method, by 0.192. However, it does not provide anything for bigram and trigram categories.

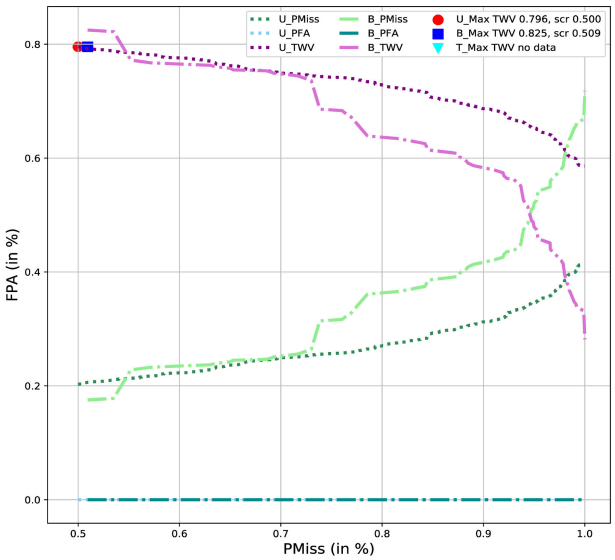


FIGURE 5. Basic query set: Term weighted threshold plot for IV words.

B. EXPERIMENT II: FULL-TEXT SEARCH

In this experiment, we assess the overall impact of incorporating a full-text search engine compared to a baseline STD system. The ASR unit generates decoded text, and we propose integrating this with the word-based STD system. This integration allows our algorithm to return documents containing words that match the input query by leveraging

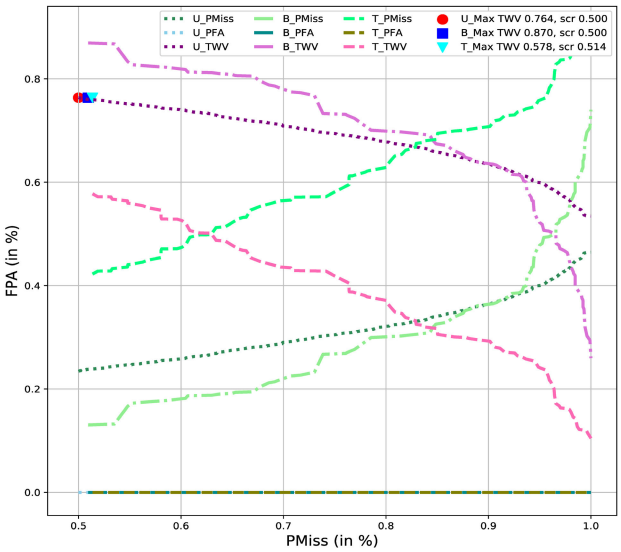


FIGURE 6. Semantic-based query set: Term weighted threshold plot for IV words.

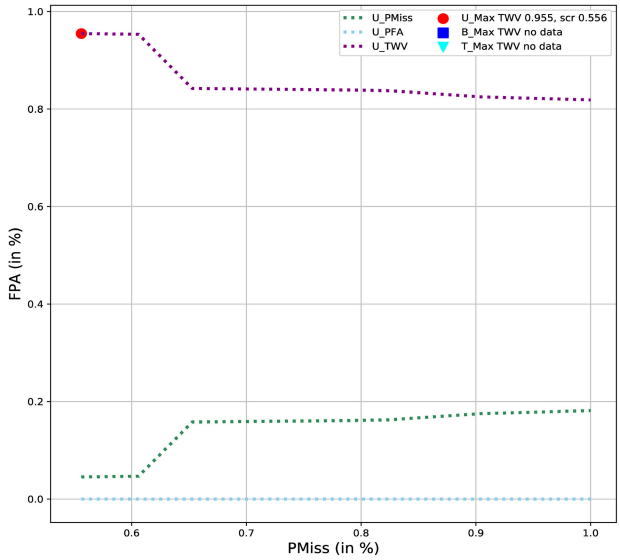


FIGURE 7. Basic query set: Term weighted threshold plot for OOV words.

features such as morphology processing and advanced full-text search syntax. Specifically, morphology processing involves lemmatization, which normalizes all found words to their base forms. Full-text search syntax enables finding query terms adjacently within any field of a document while preserving their positions. Additionally, we retrieve the spoken times of the morphology variations of keywords found in the full-text search documents using the word-based STD system. The unigram query sets, comprising 300 basic and 741 semantic-based sets, the bigram query sets with 300 basic and 2,825 semantic-based sets, and the trigram query sets consisting of 300 basic and 9,850 semantic-based sets, were passed to SphinxSearch and STD.

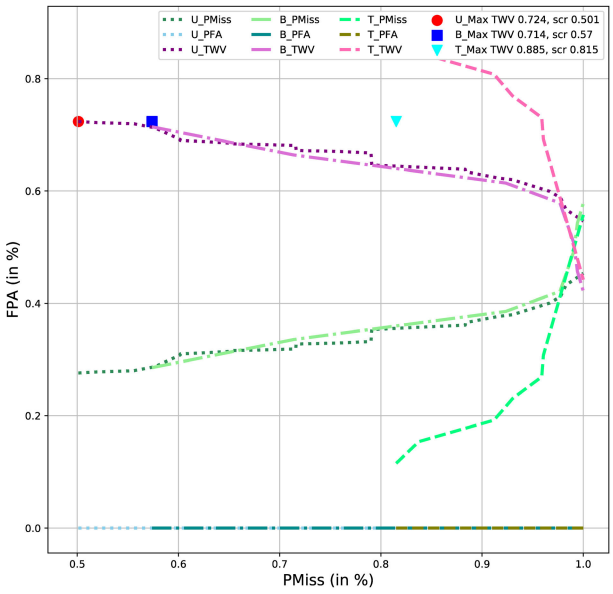


FIGURE 8. Semantic-based query set: Term weighted threshold plot for OOV words.

TABLE 4. Baseline STD vs SphinxSearch.

Query category	Basic query set		Semantically expanded query set	
	STD	SphinxSearch	STD	SphinxSearch
Unigram	9025	12679	20112	29895
Bigram	178	576	260	1570
Trigram	0	12	195	591

Table 4 presents a comparative analysis of two query types—basic and semantically expanded—across different categories: unigrams, bigrams, and trigrams, focusing on documents retrieved by both STD and SphinxSearch. The results indicate significant improvements across all query sets when integrating the full-text search engine with the word-based STD system. Notably, there is a substantial increase in document retrieval for semantic-based queries, highlighting their effectiveness in capturing a broader range of relevant results. For instance, the semantic-based unigram queries retrieved 29,895 documents using SphinxSearch, representing a 136% increase compared to the number of documents retrieved by the basic unigram queries. Similarly, in the bigram category, SphinxSearch outperformed STD by retrieving 1,570 documents for semantic-based queries—equating to a 173% increase relative to the retrieval count for the basic queries. In the trigram category, SphinxSearch retrieved 591 documents from semantic-based queries, whereas STD retrieved none using basic queries. This trend underscores the advantages of leveraging semantic expansion to enhance search relevance and document retrieval accuracy. Overall, integrating semantic enrichment into query formulation can lead to more comprehensive and relevant search results across various query types.

C. PROCESSING RESOURCE MEASUREMENTS

Fielded STD technologies are engineered to handle large volumes of data, making "speed" an essential consideration.

It is crucial for systems to record both speed and resource usage during the processing stage. These metrics help in extrapolating to larger datasets and enable fair comparisons between systems, as contrasting faster and slower systems would be unjust. The key metrics to monitor include Index Size, Indexing Speed, Search Speed, and Search Memory Usage. The proposed system was deployed on a server with the following specifications: it features a dual-socket Intel Xeon E5-2697 v4 CPU, which provides 36 cores and supports hyper-threading, resulting in a total of 72 threads. Additionally, the server is equipped with 251 GB of RAM, offering sufficient memory for efficient data processing.

The index size for word lattices in the spoken term detection system, derived from the WSJ dataset corresponding to an 80-hour collection with a total size of 29 GB and comprising 78,000 training utterances, is approximately 2.94 GB. In comparison, the index size for text transcriptions within a SphinxSearch engine is around 5 MB. Leveraging the server's capabilities, the estimated indexing time for word lattices is roughly 2 minutes, whereas the indexing time for text transcriptions is about 36 seconds. The actual Search Speed and Search Memory Usage for both word lattices and text transcriptions are as follows: for word lattices, the search speed is approximately 150 queries per second, with a search memory usage of about 150 MB. In contrast, text transcriptions achieve a higher search speed of around 800 queries per second, while their search memory usage is lower, estimated at 8 MB. We assessed our method using a set of 800 queries, which included 270 trigrams, 270 bigrams, and 260 unigrams. Within this collection, there were 40 OOV unigram queries and 30 OOV queries each for bigrams and trigrams.

Let's now explore how a single query is processed in our proposed method. Technically, our approach is capable of handling queries of any size. However, for the purpose of evaluation, we focused specifically on unigrams, bigrams, and trigrams for both IV and OOV queries. This decision was made to maintain a manageable scope for our analysis while still providing a comprehensive evaluation of our method's effectiveness. Table 5 outlines the handling of OOV trigram query based on the phrase "Origin of Parkinson's." Initially, a keyword list file is generated for the query "Origin of Parkinson's," assigning it a specific keyword ID "WSJ-9792.". This keyword list file can accommodate either a single query or a batch of queries. In the next step, an STD search is conducted using this initial keyword list. This requires an ecf file containing details of the search collection and an rttm file for scoring (Section III-B). Since the query is absent from the WSJ dataset and the STD search matches queries precisely, the STD search produces no results.

In proxy search (Section III-B.2), many of the results returned are often irrelevant due to only approximate pronunciation matches, which do not always correspond to the intended meaning of the query. In our case, the proxy search results also returned empty.

TABLE 5. An example of processing one OOV trigram query.

#	Steps of sematically expanded STD
1	**Given query (kwlist.xml):** <kw kwid="WSJ-9792"> <kwtext>Origin of Parkinson's</kwtext> </kw>
2	**STD search for given query (std_result.xml):**
3	**STD proxy search for given query (std_result_proxy.xml):**
4	**Semantically expanded query (kwlist.xml):** <kw kwid="WSJ-9793"> <kwtext>cause of parkinson</kwtext> </kw>
5	**STD search for semantically expanded query (std_result.xml):** <detected_kwlist kwid="WSJ-9793" search_time="1" oov_count="0 <kw file="001o0o2i" channel="1" tbeg="1.09" dur="1.23" score="0.990683" decision="YES" kwfile= "Doctor Langston said research into M P T P also is seeking clues to the cause of Parkinsons" /> </detected_kwlist>
6	**SphinxSearch for semantically expanded query (shx_result.xml):** <detected_kwlist kwid="WSJ-9793" search_time="1"> <kw file="001o0o2i.txt" score="0.1" decision="YES" kwfile="Doctor Langston said research into M P T P also is seeking clues to the cause of Parkinsons" /> <kw file="001o0m1f.txt" score="0.041" decision="YES" kwfile="Mr Parkinson said that agreements similar to the chip pact could probably help other U S industries battered by foreign competition" /> <kw file="001o0m1d.txt" score="0.043" decision="YES" kwfile="DRAMs are the oxygen of the computer industry Mr Parkinson claims" /> </detected_kwlist>

To enhance the search results, semantic expansion is next applied to the query (Section III-C). The query is expanded to "Cause of Parkinson," with the new keyword ID "WSJ-9793." This semantic expansion broadens the query's context, improving the system's comprehension of its intent. Both STD and SphinxSearch methods subsequently return meaningful results. In the STD search output, a relevant transcription is retrieved, indicating that the semantic expansion effectively aligned the query with the available data. For example, the phrase, "Doctor Langston said research into M P T P also is seeking clues to the cause of Parkinson's," closely matches the expanded query's intent. Similarly, the SphinxSearch (Section III-D) results reflect a successful match with the expanded query, providing additional context through various transcriptions. SphinxSearch ranks search results by assigning a relevance value to each document based on how well it matches the query. This relevance is a numerical estimate used to sort results, making the most relevant documents appear higher. The ranking process is configurable using rankers, which are specific algorithms that assign these weights. Users can adjust ranking methods, allowing for tailored search results to suit different needs. Notably, SphinxSearch assigns a high score to the retrieved result that was also detected by the STD method.

The results from STD and SphinxSearch are presented independently yet complement each other. Users can access

either set of results as needed. A primary benefit of the STD method is that it provides the start and end times of the matched query, enabling precise navigation within the audio content. Conversely, SphinxSearch offers only the document ID, indicating the source without temporal details. If users find low-weighted results in SphinxSearch, they can locate the audio file corresponding to the document ID for further review. By leveraging the features of SphinxSearch (Section III-D), documents can be retrieved where relevant words are not adjacent but separated by a few intervening words. This capability allows for searches involving relevant terms that have a few words in between. The identified substrings can then be sent to the STD search to obtain their precise timing positions within the audio, thereby enhancing the overall effectiveness of the retrieval process.

DATA AVAILABILITY

WSJ dataset that support the findings of this study are available from Linguistic Data Consortium (https://www.ldc.upenn.edu). The set of queries is available from the corresponding author, Zhanibek Kozhirbayev, upon reasonable request.

AUTHOR CONTRIBUTION

All authors have contributed equally.

VI. CONCLUSION

In this paper we proposed spoken term detection architecture built on top of the Kaldi toolkit and expanded with some features such as semantic-based query expansion and integration with full-text search engine. In order to generate queries that are semantically related to key query given by user, we used WordNet as a source of semantic information. For each given query, the system first generates all possible variations regarding the synonyms of each term in the query. Since this set of generated queries was too big and most of them are not semantically correct, we selected a small subset of these queries by applying the most frequent bigram dictionary from the Corpus of Contemporary American English. Furthermore, features of full-text search engine such as lemmatization and advanced full-text searching syntax were utilized. The former makes able to replace all found words with their normalized form, whereas the latter finds terms of query adjacently in any field in a document, but preserving their relative position within the initial query. Regarding the results of the performed experiments, the following impacts on performance of the STD system can be observed. The number of semantic alternatives of given queries plays an important role in increasing the results of the system. The generation of such alternatives is based on the lexical database for a targeted language. In our case, synonym sets in WordNet (synsets) allow us to generate lexical paraphrases but not all of them are semantically correct. Therefore, inadequate ones can be removed by using the most frequent bigram dictionary from the Corpus of Contemporary American English. Another impact is the normalization (lemmatization) of each term of given query and distribution of these terms in the spoken documents. Our experimental results show that all these approaches can be integrated to further improve performance. This study has certain limitations. The STD method is built upon the Kaldi toolkit, and for our experiments, we only used the WSJ dataset, which contains approximately 80 hours of training data. Furthermore, we did not employ deep learning methods in our approach. Additionally, while the semantic expansion technique increases the number of documents retrieved, it may also introduce noise and potentially irrelevant results. This phenomenon arises because the expanded queries can lead to results that, although semantically related, do not align closely with the user's original intent. Acknowledging this trade-off is important, as it highlights the need for further investigation into filtering and ranking mechanisms that can enhance the relevance of the expanded results while retaining the benefits of increased retrieval. Addressing these limitations will be a focus of future work, where we plan to utilize larger and more diverse datasets, explore deep learning-based approaches, and extend our research to include neural methods for generating semantically related alternatives of queried phrases, while also considering the implications of relevance in the search results.

REFERENCES

[1] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 416–421.

[2] D. Karakos and R. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *Proc. Interspeech*, Sep. 2014, pp. 2469–2473.

[3] C. van Heerden, D. Karakos, K. Narasimhan, M. Davel, and R. Schwartz, "Constructing sub-word units for spoken term detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5780–5784.

[4] Y. He, P. Baumann, H. Fang, B. Hutchinson, A. Jaech, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Using pronunciation-based morphological subword units to improve OOV handling in keyword search," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 1, pp. 79–92, Jan. 2016.

[5] M. Saraclar, A. Sethy, B. Ramabhadran, L. Mangu, J. Cui, X. Cui, B. Kingsbury, and J. Mamou, "An empirical study of confusion modeling in keyword search for low resource languages," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 464–469.

[6] D. Karakos and R. M. Schwartz, "Combination of search techniques for improved spotting of OOV keywords," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5336–5340.

[7] H. Su, V. T. Pham, Y. He, and J. Hieronymus, "Improvements on transducing syllable lattice to word lattice for keyword search," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4729–4733.

[8] L. Mangu, B. Kingsbury, H. Soltau, H.-K. Kuo, and M. Picheny, "Efficient spoken term detection using confusion networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 7844–7848.

[9] Z. Kozhirbayev, M. Karabalayeva, and Z. Yessenbayev, "Spoken term detection for Kazakh language," in *Proc. The 4-th Int. Conf. Comput. Process. Turkic Lang.*, 2016, pp. 47–52.

[10] X. Wang, P. Zhang, X. Na, J. Pan, and Y. Yan, "Handling OOVWords in Mandarin spoken term detection with an hierarchical n-gram language model," *Chin. J. Electron.*, vol. 26, no. 6, pp. 1239–1244, Nov. 2017.

[11] P. Karanasou, L. Burget, D. Vergyri, M. Akbacak, and A. Mandal, "Discriminatively trained phoneme confusion model for keyword spotting," in *Proc. Interspeech*, Sep. 2012, pp. 2434–2437.

[12] L.-S. Lee, J. Glass, H.-Y. Lee, and C.-A. Chan, "Spoken content retrieval—Beyond cascading speech recognition with text retrieval," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 9, pp. 1389–1420, Sep. 2015.

[13] D. Kaneko, R. Konno, K. Kojima, K. Tanaka, S.-W. Lee, and Y. Itoh, "Constructing acoustic distances between subwords and states obtained from a deep neural network for spoken term detection," in *Proc. Interspeech*, Aug. 2017, pp. 2879–2883.

[14] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi OpenKWS system: Improving low resource keyword search," in *Proc. Interspeech*, Aug. 2017, pp. 3597–3601.

[15] J. Švec, L. Šmídl, and J. V. Psutka, "An analysis of the RNN-based spoken term detection training," in *Proc. Int. Conf. Speech Comput.*, 2017, pp. 119–129.

[16] A. Ito and M. Koizumi, "Spoken term detection of zero-resource language using machine learning," in *Proc. Int. Conf. Intell. Inf. Technol.*, Feb. 2018, pp. 45–49.

[17] P. Golik, Z. Tüske, K. Irie, E. Beck, R. Schlüter, and H. Ney, "The 2016 RWTH keyword search system for low-resource languages," in *Proc. Int. Conf. Speech Comput.*, 2016, pp. 719–730.

[18] J. Singh, A. Sharan, and M. Saini, "Term co-occurrence and context window-based combined approach for query expansion with the semantic notion of terms," *Int. J. Web Sci.*, vol. 3, no. 1, p. 32, 2017.

[19] V. M. Ngo, T. H. Cao, and T. M. V. Le, "WordNet-based information retrieval using common hypernyms and combined features," 2018, *arXiv:1807.05574*.

[20] H. Kamper, G. Shakhnarovich, and K. Livescu, "Semantic speech retrieval with a visually grounded model of untranscribed speech," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, May 2018, pp. 2514–2517.

[21] R. Qu, Y. Fang, W. Bai, and Y. Jiang, "Computing semantic similarity based on novel models of semantic representation using Wikipedia," *Inf. Process. Manage.*, vol. 54, no. 6, pp. 1002–1021, Nov. 2018.

[22] Y. Jiang, W. Bai, X. Zhang, and J. Hu, "Wikipedia-based information content and semantic similarity computation," *Inf. Process. Manage.*, vol. 53, no. 1, pp. 248–265, Jan. 2017.

[23] P. Arora, J. Foster, and G. J. Jones, "Query expansion for sentence retrieval using pseudo relevance feedback and word embedding," in *Proc. Int. Conf. Cross-Language Eval. Forum Eur. Lang.*, 2017, pp. 97–103.

[24] J. Tekli, R. Chbeir, A. J. M. Traina, C. Traina, K. Yetongnon, C. R. Ibanez, M. Al Assad, and C. Kallas, "Full-fledged semantic indexing and querying model designed for seamless integration in legacy RDBMS," *Data Knowl. Eng.*, vol. 117, pp. 133–173, Sep. 2018.

[25] S. Jiang, T. F. Hagelien, M. Natvig, and J. Li, "Ontology-based semantic search for open government data," in *Proc. IEEE 13th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2019, pp. 7–15.

[26] D.-H. Ngo, M. Kemp, D. Truran, B. Koopman, and A. Metke-Jimenez, "Semantic search for large scale clinical ontologies," in *AMIA Annu. Symp. Proc.*, 2021, p. 910.

[27] *Video Search and Discovery*. Accessed: Nov. 4, 2024. [Online]. Available: https://www.panopto.com/features/video-search

[28] *The Microsoft Audio Video Indexing Service*. Accessed: Nov. 4, 2024. [Online]. Available: https://www.microsoft.com/en-us/research/project/mavis

[29] D. O'Shaughnessy, "Automatic speech recognition," in *Proc. CHILEAN Conf. Electr., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Oct. 2015, pp. 417–424. Accessed: 2024-11-04.

[30] G. Li, H. Zhu, G. Cheng, K. Thambiratnam, B. Chitsaz, D. Yu, and F. Seide, "Context-dependent deep neural networks for audio indexing of real-life data," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2012, pp. 143–148.

[31] A. Guo, A. Faria, and K. Riedhammer, "Remeeting-deep insights to conversations," in *Proc. INTERSPEECH*, 2016, pp. 1964–1965.

[32] A. Faria and K. Riedhammer, "Remeeting get more out of meetings," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1964–1965.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, May 2011, pp. 1–23.

[34] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. 2nd Int. Conf. Spoken Lang. Process. (ICSLP)*, Oct. 1992, pp. 899–902.

[35] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 8, pp. 2338–2347, Nov. 2011.

[36] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, Aug. 2007, pp. 314–317.

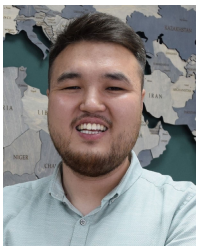
[37] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[38] M. Davies, "The 385+ million word corpus of contemporary American English (1990–2008+): Design, architecture, and linguistic insights," *Int. J. Corpus Linguistics*, vol. 14, no. 2, pp. 159–190, Jun. 2009.

[39] C. C. Aggarwal, "Information retrieval and search engines," in *Machine Learning for Text*. Cham, Switzerland: Springer, 2018, pp. 259–304.

[40] *Top Frequent Google Searches*. Accessed: Apr. 30, 2024. [Online]. Available: http://trends.google.com/trends

[41] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," *Proc. SIGIR*, vol. 7, pp. 51–57, May 2006.



ZHANIBEK KOZHIRBAYEV received the master's degree in information technology (distributed computing) from the University of Melbourne, Australia, in 2015, and the Ph.D. degree in computer science from L. N. Gumilyov Eurasian National University (ENU), Kazakhstan, in 2019. Currently, he is a Senior Researcher with the National Laboratory Astana, Nazarbayev University. His research interests include natural language processing, speech recognition and synthesis, artificial intelligence, and cloud computing.

ZHANDOS YESSENBAYEV received the Specialist degree in applied mathematics from Moscow Institute of Steel and Alloys (MISIS), Russia, in 2006, the master's degree in computer science from The University of Chicago, USA, in 2008, and the Ph.D. degree in computer science from L. N. Gumilyov Eurasian National University (ENU), Kazakhstan. Currently, he is a Subject Content Coordinator with the Computer Science Department, BINUS University International Program. His research interests include artificial intelligence, computational linguistics, natural language processing, speech recognition and synthesis, acoustic phonetics, and computational topology.