Written Term Detection Improves Spoken Term Detection

Bolaji Yusuf, *Graduate Student Member, IEEE*, and Murat Saraçlar, *Senior Member, IEEE*

***Abstract*—End-to-end (E2E) approaches to keyword search (KWS) are considerably simpler in terms of training and indexing complexity when compared to approaches which use the output of automatic speech recognition (ASR) systems. This simplification however has drawbacks due to the loss of modularity. In particular, where ASR-based KWS systems can benefit from external unpaired text via a language model, current formulations of E2E KWS systems have no such mechanism. Therefore, in this paper, we propose a multitask training objective which allows unpaired text to be integrated into E2E KWS without complicating indexing and search. In addition to training an E2E KWS model to retrieve text queries from spoken documents, we jointly train it to retrieve text queries from masked written documents. We show empirically that this approach can effectively leverage unpaired text for KWS, with significant improvements in search performance across a wide variety of languages. We conduct analysis which indicates that these improvements are achieved because the proposed method improves document representations for words in the unpaired text. Finally, we show that the proposed method can be used for domain adaptation in settings where in-domain paired data is scarce or nonexistent.**

***Index Terms*—Keyword search, spoken term detection, keyword spotting, end-to-end keyword search, multitask learning, domain adaptation, masked language modeling.**

I. Introduction

KEYWORD search (KWS), known alternatively as spoken term detection, is a branch of spoken content retrieval task concerned with retrieving speech segments where a user-provided query is uttered. Given a user's short written query, a KWS system searches an archive of speech and returns those utterances in the archive hypothesized to contain the query, timestamps showing the exact location of each hypothesis and a set of scores denoting the system's confidence in the hypotheses.

Manuscript received 5 November 2023; revised 11 April 2024; accepted 13 May 2024. Date of current version 26 June 2024. This work was supported in part by the Czech Ministry of Interior under Project VJ01010108 "ROZKAZ". Computing on IT4I supercomputer was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90254). Computing on the ROYAL compute server was supported by the Turkish Directorate of Strategy and Budget under the ROYAL Project CB SBB 2019K12-149250. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Samuel Thomas. *(Corresponding author: Bolaji Yusuf.)*

Bolaji Yusuf is with the Boğaziçi University Department of Electrical and Electronics Engineering, 34342 Istanbul, Türkiye, and also with the Brno University of Technology, Faculty of Information Technology, Speech@FIT, Brno 612 00, Czechia (e-mail: bolaji.yusuf@boun.edu.tr).

Murat Saraçlar is with the Boğaziçi University Department of Electrical and Electronics Engineering, 34342 Istanbul, Türkiye (e-mail: muratsaraclar@ieee.org).

Digital Object Identifier 10.1109/TASLP.2024.3407476

The traditional approach to KWS involves building a large vocabulary continuous speech recognition (LVCSR) system, using it to decode the archive, and, from the resulting lattices, constructing an inverted index in which queries are searched [1], [2], [3], [4], [5]. However, this approach inherits the shortcomings of the underlying automatic speech recognition (ASR), most notably, the non-trivial complexity and computational costs associated with ASR training and decoding. There has therefore been interest in end-to-end (E2E) approaches which eschew the ASR part of the KWS pipeline. These E2E systems are trained to directly predict whether, and where, a query occurs in a given speech segment, leading to a much simpler system in terms of training and search [6], [7], [8]. Although E2E KWS systems, like the one in this paper, still rely on simple ASR systems to get timing information at training time, they feature a much more simplified indexing and search scheme, comparable in complexity to acoustic modeling in ASR.

Its complexities notwithstanding, ASR-based KWS still maintains some advantages over E2E KWS in terms of both efficiency and accuracy of search. While ASR-based systems transcribe the archives into text-based structures such as factor transducers [4], confusion networks [9] and position specific posterior lattices [10] which allow fast, sub-linear indexes, E2E methods generally rely on inner-product search with fixed frame-rate vector document representations. Thus, the storage and computational cost of E2E KWS grows linearly in the duration of the archive. In addition to efficiency, ASR-based KWS also outperforms E2E KWS in terms of search accuracy. The performance advantage of ASR-based KWS is especially pronounced for short queries while E2E KWS tend to have the advantage for longer queries [11]. Nevertheless, the two approaches tend to be complimentary and prior work has achieved significant improvements in search accuracy by combining them across queries of all lengths [7], [8], [11].

As with E2E systems in other domains, the simplification in E2E KWS comes at the expense of data efficiency as these systems generally require larger amounts of labeled training data than their more modular counterparts. Of particular interest to us is that ASR-based systems (even end-to-end ASR systems) can be improved with unpaired text data independent of the paired training speech-text data. This naturally raises the question of how to use large text-only corpora to improve E2E KWS systems. Since E2E KWS systems, as have been explored in literature, model span probabilities and not word probabilities, they cannot make use of language models which constitute the primary method of using text-only data to improve ASR

systems. On the other hand, there has been a recent trend in E2E ASR of using joint training with text-to-text transduction tasks to integrate the unpaired text into ASR training and reduce the dependency on external language models during ASR inference [12], [13], [14].

Inspired by these approaches, in this paper, we propose training an E2E KWS system jointly with an auxiliary text-to-text task. Taking the E2E KWS model of [11] as the baseline, we introduce a joint training scheme where, in addition to the baseline training of predicting the locations of short written queries in *speech* segments, the model is also trained to predict the locations of written queries in masked *written* sentences. As this auxiliary training objective can be computed with purely textual inputs, it provides a way to incorporate text-only corpora into the KWS model.

We conduct extensive experiments which yield the following results:

- The proposed model consistently and significantly improves keyword search performance across several languages, domains and input feature choices. Moreover, the proposed joint speech-text training scheme is orthogonal to multilingual pretraining and data augmentation, and can be used alongside them to achieve even better performance.
- The proposed joint training method improves document representation of phrases contained in the auxiliary unpaired text, both when such phrases exist in spoken form in the paired KWS training data and when they do not.
- Training with unpaired text from a domain improves performance on test sets in that domain, and therefore provides a viable solution for dealing with domain mismatches between the KWS train and test sets.

The rest of the paper is organized as follows: Section II covers previous related work; Section III recapitulates the baseline end-to-end KWS framework which we build upon and then describes the proposed model; Section IV details the experiments conducted and discusses the results of those experiments; Section V concludes the paper with a summary and future research directions.

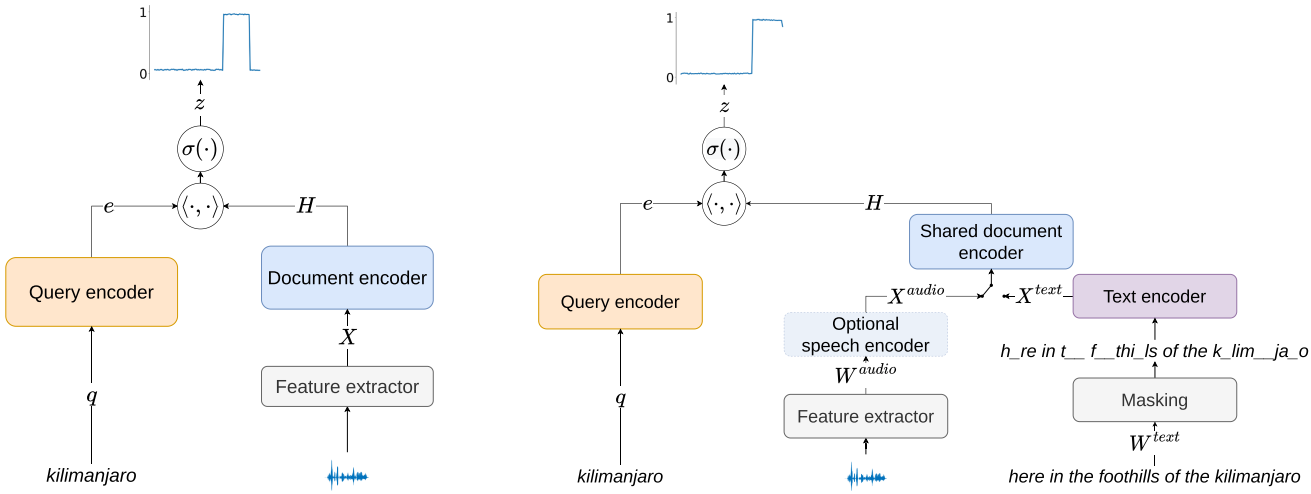
II. Related Work

Our work falls within the gamut of ASR-free KWS systems which attempt to simplify the KWS pipeline. Some of the earlier approaches to this include the use of point-process models [15], [16] and dynamic time warping [17], [18], [19]. More recent approaches use neural architectures which encode queries and documents and effect search by combining those encodings [6], [7], [8], [20], with especially [7] and [8] achieving high efficiency by using completely separated encoders for the query and the document and combining the encodings by simple dot-products. Several improvements have been made to the document representations including pretraining the document encoder as an autoencoder [6], an ASR encoder [21], a self-supervised model [22] and a multilingual KWS document encoder [11]. However, none of these approaches have been able to integrate unpaired text directly into the keyword search model. We note that [6] and [20] pretrain their *query* encoders using external text. Unlike those, we use the unpaired text to better train our *document* encoder. As we will show in Section IV-E, using unpaired text to train the document encoder with our method leads to significantly better KWS performance than using it for training only the query encoder.

Classical ASR-based methods can easily incorporate unpaired text as they are modular. Various works have shown that using external text can significantly improve ASR-based KWS by using such text to augment the ASR pronunciation lexicon [23], [24] and the language model [25], [26]. These works show that the KWS improvements can be substantial even when improvements to the underlying ASR system are less pronounced, due to the effect on rare and out-of-vocabulary (OOV) queries. The fundamental question of this paper is how to leverage such external text for end-to-end KWS methods which possess neither lexicons nor language models.

A related line of research is the use of unpaired speech for training. This includes pretraining with surrogate unsupervised objectives on large, untranscribed corpora and then finetuning on paired data [27], [28], [29], [30], or semi-supervised training which involves training a seed ASR model on small transcribed data, using it to transcribe otherwise unlabeled data and then adding the resulting automatically-transcribed data into the training pool for further training [31], [32], [33]. The work most related to ours in this direction is [8], in which an ASR system was used to transcribe large quantities of speech for training E2E KWS. Overall, these works, which improve performance by making use of unpaired speech, are orthogonal to ours which makes use of unpaired text. In our experiments, we use the pretrained model from [30] to extract input features and we show that adding unpaired text with our method yields consistent improvements.

A more related line of work involves using unpaired text data directly in ASR training. One way of doing so is using a text-to-speech (TTS) system to generate matching speech, and including the resulting paired data as part of ASR training [34], [35], [36]. However TTS adds its own significant computational and modeling complexity. Moreover, robust TTS systems are generally a luxury only available for high resource languages. Therefore, joint speech-text models such as MMDA [37], PSDA [38], MUTE-L [12], USTED [13], Textogram [14] and MAESTRO [39] have gained interest as a way of integrating unpaired text into end-to-end ASR to improve performance on ASR, as well as other downstream tasks such as spoken language understanding [40] and spoken machine translation [41]. These models incorporate unpaired text by treating the entire ASR model as part of a larger multimodal text generator, some of whose parameters can be jointly trained for text-to-text transduction without any explicit TTS synthesis. By improving the underlying ASR system, these methods can plausibly be used to improve ASR-based KWS systems, especially recently proposed KWS systems based on end-to-end ASR [42], [43], [44], [45]. However, they cannot work for end-to-end ASR-free KWS systems which are generally discriminators rather than text generators. Our proposed method introduces a surrogate objective for incorporating text into the discriminative framework ASR-free KWS systems.



(a) BeKWS model. (b) Proposed JOSTER model.

Fig. 1. Illustrations of the baseline and proposed systems. Both accept a written query and a spoken document and return a sequence of probabilities indicating where, if anywhere, in the document the query is spoken. The proposed system, however, also accepts documents in text form through a text encoder, thereby allowing the possibility of training with text-only data.

III. Methods

In this section, we describe the joint training method that we propose for KWS. First, in Section III-A, we recapitulate the baseline E2E KWS method from [11]—which we will subsequently refer to as baseline end-to-end KWS (BeKWS)—as it forms the basis of our method. Then in Section III-B, we describe the proposed joint model, which we will subsequently refer to as the joint speech and text retriever (JOSTER). In Section III-C, we provide details on how we train the model.[1]

A. Baseline End-to-End Keyword Search (BeKWS)

BeKWS—depicted in Fig. 1(a)—is a model trained to predict the probabilities of a query occurring in each frame of a spoken document. For a (possibly multi-word) query $\mathbf{q} = (q_1, q_2, \ldots, q_K)$ comprising a sequence of K letters and a document $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ comprising a sequence of N acoustic frames, the model is used to predict the sequence $\mathbf{y}(\mathbf{q}, \mathbf{X}) = (y_1, \ldots, y_N)$, where each $y_n \in \{0, 1\}$ is a binary random variable indicating the existence of the query, i.e:

$$y_n = \begin{cases} 1, & \text{if } \mathbf{q} \text{ is spoken in } \mathbf{X} \text{ in a time span including } n \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The model comprises a document encoder and a query with parameters $\boldsymbol{\Delta}$ and $\boldsymbol{\psi}$ respectively. Given the query $\mathbf{q}$ and the document $\mathbf{X}$, the model outputs the sequence $\mathbf{z}(\mathbf{q}, \mathbf{X}; \boldsymbol{\theta}) = (z_1, \ldots, z_N)$ of query occurrence probabilities where:

$$z_n(\mathbf{q}, \mathbf{X}; \boldsymbol{\theta}) = \sigma\left(\mathbf{h}_n^\top \mathbf{e}(\mathbf{q}; \boldsymbol{\psi})\right), \tag{2}$$

where $\boldsymbol{\theta} := \{\boldsymbol{\Delta}, \boldsymbol{\psi}\}$; $\mathbf{h}_n$ is the nth frame of $\mathbf{H}(\mathbf{X}; \boldsymbol{\Delta})$, a downsampled representation of $\mathbf{X}$ computed by the document encoder; $\mathbf{e}(\mathbf{q}; \boldsymbol{\psi})$ is the vector representation of the query computed by the query encoder, and $\sigma(\cdot)$ is the logistic sigmoid function. Thus, $z_n(\mathbf{q}, \mathbf{X}; \boldsymbol{\theta})$ is interpreted as $P_{\boldsymbol{\theta}}(y_n = 1 | \mathbf{q}, \mathbf{X})$.

Given a training dataset containing of a set of spoken documents $\mathcal{X}$, the model is trained by stochastic gradient descent to minimize the negative log-likelihood of the indicators:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{\mathbf{q} \in Q} \sum_{\mathbf{X} \in \mathcal{X}} \sum_{n} -\log P_{\boldsymbol{\theta}}(y_n | \mathbf{q}, \mathbf{X}), \tag{3}$$

where the training queries are taken from $\mathcal{Q}$, the set of all unigrams, bigrams, trigrams in the transcripts of $\mathcal{X}$, and the word-level timestamps required for training are obtained by forced-alignment with an HMM-GMM-based ASR system trained on the KWS training data. In practice, the model is trained with a modified cross-entropy objective which was introduced in [7] (and which we will recap in Section III-C) because it has been shown to outperform the vanilla binary cross-entropy implied by (3).

B. Joint Speech and Text Retriever (JOSTER)

The approach we propose for incorporating unpaired text into E2E KWS, JOSTER—depicted in Fig. 1(b)—involves modifying the document encoder of BeKWS to accept not just acoustic inputs but also textual ones. To do so, we introduce a pair of modality encoders which transform input from their respective modalities into a shared space; a speech-only encoder which takes spoken documents as input, and a text-only encoder which takes written sentences as input. The output of either encoder can then be fed into a shared document encoder, and combined as in (2) with the output of the (shared) query encoder to obtain

[1]Code for BeKWS and JOSTER available at https://github.com/bolajiy/golden-retriever

probabilities of occurrence of the query in either spoken or written sentences.

For a spoken document, $\mathbf{W}^{\text{audio}} = (\mathbf{w}_1, \ldots, \mathbf{w}_N)$, we compute its representation $\mathbf{X}^{\text{audio}}(\mathbf{W}^{\text{audio}}; \boldsymbol{\Delta}_{\text{audio}})$ by passing it through an optional speech-only encoder with parameters $\boldsymbol{\Delta}_{\text{audio}}$. Henceforth, to reduce clutter, we drop the functional form $\mathbf{X}^{\text{audio}}(\mathbf{W}^{\text{audio}}; \boldsymbol{\Delta}_{\text{audio}})$, and simply write $\mathbf{X}^{\text{audio}}$ with the understanding that the dependency is implied. Note that, with the change of the model, we have had to make a slight change in notation: in Section III-A, $\mathbf{X}$ denoted both the sequence of acoustic features and the document encoder input (since these are identical for BeKWS); here, $\mathbf{W}^{\text{audio}}$ denotes the sequence of acoustic features, while $\mathbf{X}^{\text{audio}}$ refers to the document encoder input which computed on $\mathbf{W}^{\text{audio}}$. For most of our experiments, we do not use a speech-only encoder at all. Rather, we use the text-only encoder to project written documents to the space of acoustic features, i.e., by default, $\boldsymbol{\Delta}_{\text{audio}} = \varnothing$ and $\mathbf{X}^{\text{audio}} = \mathbf{W}^{\text{audio}}$.

To compute the representation for a written document, $\mathbf{W}^{\text{text}} = (w_1, \ldots, w_N)$, we first mask it to obtain $\tilde{\mathbf{W}}^{\text{text}} = (\tilde{w}_1, \ldots, \tilde{w}_N)$:

$$\tilde{w}_n = \begin{cases} \_, & \text{with probability } \pi, \\ w_n, & \text{with probability } 1 - \pi, \end{cases} \tag{4}$$

where _ is a special mask symbol. Then we incorporate a rudimentary duration model transforming the input to $\hat{\mathbf{W}}^{\text{text}} = (\hat{w}_1, \ldots, \hat{w}_N)$ where each $\hat{w}_n$ is obtained by simply repeating $\tilde{w}_n$ ρ times. For instance, the phrase $\mathbf{W}^{\text{text}} = \texttt{thecat}$ might be converted to $\tilde{\mathbf{W}}^{\text{text}} = \texttt{t\_ec\_t}$, and then, if $\rho = 2$, to $\hat{\mathbf{W}}^{\text{text}} = \texttt{tt\_\_eecc\_\_tt}$. This final representation is then input into the text encoder—a neural network with an embedding lookup input layer—with parameters $\boldsymbol{\Delta}_{\text{text}}$ to obtain $\mathbf{X}^{\text{text}}(\mathbf{W}^{\text{text}}; \boldsymbol{\Delta}_{\text{text}})$. We determine the values of π and ρ experimentally (see Section IV-C for an analysis of their impact).

Having obtained the modality-specific representations, we can use (2) to get the occurrence probabilities $P_{\boldsymbol{\theta}}(y_n|\mathbf{q}, \mathbf{X}^{\text{audio}})$ or $P_{\boldsymbol{\theta}}(y_n|\mathbf{q}, \mathbf{X}^{\text{text}})$ for any query $\mathbf{q}$ where the parameters $\{\boldsymbol{\Delta}, \boldsymbol{\psi}\}$ are shared by both the speech-text retrieval and the text-text retrieval.

As stated in Section III-A, for spoken documents, $\mathbf{y}(\mathbf{q}, \mathbf{X}^{\text{audio}})$ is defined by whether the query is spoken at a time span of the document. For written documents, $\mathbf{y}(\mathbf{q}, \mathbf{X}^{\text{text}})$ is defined by whether the query occurs exactly at a given location. Using the example from above, with document sentence $\mathbf{W}^{\text{text}} = \texttt{thecat}$ and $\hat{\mathbf{W}}^{\text{text}} = \texttt{tt\_\_eecc\_\_tt}$, and a query $\mathbf{q} = \texttt{cat}$,

$$\mathbf{y}\left(\mathbf{q}, \mathbf{X}^{\text{text}}\right) = 000000111111. \tag{5}$$

C. Training

We train the model jointly on a paired speech-text dataset, $\mathcal{X}^{\text{audio}}$, and an unpaired text-only one, $\mathcal{X}^{\text{text}}$, using stochastic gradient descent. At each training step, k, we sample the dataset μ uniformly from $\{\text{audio}, \text{text}\}$ and minimize:

$$J_k^{\mu} = \sum_{l=1}^{L} \sum_{m=1}^{M} f\left(\mathbf{z}\left(\mathbf{q}_{k,l}^{\mu}, \mathbf{X}^{\mu}\left(\mathbf{W}_{k,l,m}^{\mu}; \boldsymbol{\Delta}_{\mu}\right); \boldsymbol{\theta}\right)\right.,$$

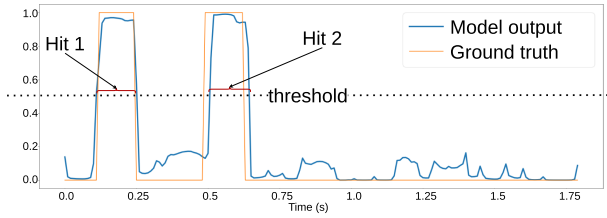


Fig. 2. Post-processing of model output into KWS hypotheses. Contiguous regions with scores above 0.5 are selected as hits, and the confidence of each hit is the median score in corresponding region.

$$\left.\mathbf{y}\left(\mathbf{q}_{k,l}^{\mu}, \mathbf{X}^{\mu}\left(\mathbf{W}_{k,l,m}^{\mu}; \boldsymbol{\Delta}_{\mu}\right)\right)\right), \tag{6}$$

where $\{\mathbf{q}_{k,1}^{\mu} \ldots \mathbf{q}_{k,L}^{\mu}\}$ is a mini-batch of L queries sampled randomly from the set of unigrams, bigrams and trigrams of the dataset $\mathcal{X}^{\mu}$; [2] $\{\mathbf{W}_{k,l,1}^{\mu}, \ldots, \mathbf{W}_{k,l,M}^{\mu}\}$ is a set of documents sampled from the dataset such that $\mathbf{W}_{k,l,1}^{\mu}$ contains $\mathbf{q}_{k,l}^{\mu}$ while the other $M-1$ documents are sampled randomly; $\mathbf{X}^{\mu}(\cdot)$ is the output of the corresponding modality-specific encoder; $\mathbf{z}(\cdot)$ is the model output as described by (2); $\mathbf{y}(\cdot)$ is the ground truth as described by (1) and (5); and $f(\cdot)$ is the modified binary cross-entropy function defined as:

$$f(z, y) = -\sum_{n=1}^{N} \Big(\mathbb{1}_{z_n > 1-\phi} \cdot (1 - y_n) \log(1 - z_n) + \mathbb{1}_{z_n < \phi} \cdot \lambda \cdot y_n \log z_n \Big), \tag{7}$$

where ϕ is a hyper-parameter controlling the tolerance of the objective to easily-classified frames and λ controls the relative weighting of positive to negative frames.

D. Post-Processing for Keyword Search

After the model is trained, we no longer require the text-only document encoder ($\boldsymbol{\Delta}^{\text{text}}$), i.e., at search time, JOSTER becomes effectively identical to BeKWS. We post-process the output of the query and spoken document encoders for KWS using the procedure illustrated in Fig. 2. As in BeKWS, for a given document, the query is detected if there exists "islands" of consecutive frames whose sigmoid outputs, $P_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{q}, \mathbf{X}^{audio})$, exceed some threshold. We set this threshold to 0.5 in all our experiments, although we found search performance to be stable for thresholds between 0.4 and 0.7. The first and last frame of the sequence are taken as the timestamps of the query *hit* and the median probability of the sequence is taken as the confidence. Finally, we discard hits which are shorter than 40 ms $\times$ query length in letters.

IV. Experiments

In this section, we conduct experiments to analyze various aspects of the proposed JOSTER model. First, we describe the

[2] Note that, as in the baseline, we consider multiple occurrences of the same training query to be distinct elements of the set, so that the probability of sampling a particular training query is directly proportional to the number of times it occurs in the training data.

TABLE I
STATISTICS OF THE TRAINING TEXT CORPORA

Language	Vocab-L	Vocab-F	OOV-L (%)	OOV-F (%)
Assamese	7661	22033	12.6	1.6
Bengali	7933	24339	13.1	2.1
Pashto	6186	17640	11.4	2.3
Turkish	10110	38311	19.2	6.5
Zulu	13764	54295	20.3	6.7

Vocab-L denotes the vocabulary size of the LLP (paired) training data, Vocab-F denotes the vocabulary size of the FLP (unpaired) training data, OOV-L and OOV-F refer to the proportion of evaluation queries in each language which are out-ofvocabulary of the LLP and FLP training data respectively.

experiment setup including datasets, input features, metrics and model configuration. Next we present a macro comparison of the KWS performance of the proposed method to that of the BeKWS baseline. Then we analyze the effect of the text representation hyperparameters on search performance. Afterwards, we conduct experiments to understand how JOSTER achieves its improvements with analyses of the effect of the size and choice of unpaired text, the performance difference on various kinds of queries and, finally, the effect of the domain of the unpaired text on KWS performance.

A. Experimental Setup

1) Datasets: We conduct the bulk of our experiments on the IARPA Babel corpora for low resource ASR and KWS,[3] from which we select Assamese,[4] Bengali,[5] Pashto,[6] Turkish[7] and Zulu[8] as the target languages for KWS training and testing.

For each language, we use the limited language pack (LLP) subset which contains about 10-hours of training data per language as the paired training data. We use the text from the full language packs (FLP) as the unpaired text for each language. These contain 5-6 times as many sentences as the LLP subset.

Each language has a 10-hour development (dev) set and a 5-hour evaluation (eval) set, with a few thousand queries per set. Table I gives a summary of the text data for each language including the size of the paired text lexicon, the size of the unpaired text lexicon, and the proportion of evaluation queries which are OOV with respect to each text source. We note that Turkish and Zulu, both agglutinative languages, have larger vocabulary sizes and higher OOV rates.

In addition to these, we use the LLP data from 19 other languages of the Babel corpus (about 190 hours in total) for multilingual pretraining of the KWS model—which was shown to significantly improve KWS performance for BeKWS in [11]–in order to measure whether and how well the proposed method can be used to improve a multilingually pretrained KWS baseline.

2) Acoustic Features: We use features from a pretrained 300 million parameter XLS-R model [30] as the acoustic input to our KWS system.[9] In a preliminary experiment on the Turkish development set, we tried using the outputs of 5th, 10th, 15th, 20th and 23 rd (final) layers of the XLS-R model and found that the 15th layer worked best, and so we use it forsubsequent experiments. Note that, due to computational constraints, we only use the XLS-R model as a feature extractor rather than finetune it.

In addition to the 1024-dimensional XLS-R features, we also consider 42-dimensional multilingual bottleneck features (BNF) in Section IV-B as an alternative acoustic input, giving us yet another axis along which to analyze the proposed method's performance. The BNF extractor is a TDNN-based [46] multilingual acoustic model which we trained in block-softmax fashion [47] to classify clustered context-dependent triphone states on the other Babel languages' LLP data.

3) Metric: We report the term weighted values (TWV) in all our experiments [48], which is a measure of weighted recall and precision averaged across queries. The TWV of a set of queries $\mathcal{Q}$ at a threshold ζ is defined as:

$$\mathrm{TWV}\left(\zeta, \mathcal{Q}\right) = 1 - \frac{1}{\mathcal{Q}} \sum_{q \in \mathcal{Q}} \left(P_{\mathrm{miss}}\left(q, \zeta\right) + \beta P_{\mathrm{FA}}\left(q, \zeta\right) \right), \quad (8)$$

where $P_{\mathrm{miss}}(q, \zeta)$ is the probability of misses, $P_{\mathrm{FA}}(q, \zeta)$ is the probability of false alarms and β is a parameter which controls the relative importance of the two. Following prior NIST evaluations [49], [50], we set $\beta = 999.9$. The threshold ζ is tuned on the dev sets. For the dev sets, we report the maximum term weighted value (MTWV) which is the TWV at the threshold which maximizes it. For the eval sets, we report the actual term weighted value (ATWV) which is computed by using the threshold tuned on the dev set. Note that we report all our TWV in percentages, i.e., we always multiply the TWVs as defined in (8) by 100, so 100% corresponds to a perfect system with no misses and false alarms, whereas 0% corresponds to a system with no outputs, and negative TWV (up to $-\beta \times 100\%$) is possible for systems with a preponderance of false alarms.

In addition to the TWV, we report Detection Error Tradeoff (DET) curves in Section IV-B. The DET curves show a plot of miss probabilities vs false alarm probabilities for a KWS system, giving a more holistic view of keyword search performance. KWS systems with DET curves closer to the lower-left corner of the plot have better false alarm to miss tradeoffs and are thus considered better. We use NIST's F4DE toolkit[10] for computing TWVs and generating DET plots.

We adopt keyword-specific thresholding for across-query score normalization [51] in order to allow various queries with different score distributions to be compared with a single global threshold.

4) Model Configuration and Hyper-Parameters: We base the architecture of our model on [11]. The query encoder is a network with a 32-dimensional embedding layer for computing vector representations of each input grapheme, followed by 2 bidirectional gated recurrent unit (GRU) layers with 256 output

[3][Online]. Available: https://www.iarpa.gov/index.php/research-programs/babel
[4][Online]. Available: https://catalog.ldc.upenn.edu/LDC2016S06
[5][Online]. Available: https://catalog.ldc.upenn.edu/LDC2016S08
[6][Online]. Available: https://catalog.ldc.upenn.edu/LDC2016S09
[7][Online]. Available: https://catalog.ldc.upenn.edu/LDC2016S10
[8][Online]. Available: https://catalog.ldc.upenn.edu/LDC2017S19

[9][Online]. Available: https://dl.fbaipublicfiles.com/fairseq/wav2vec/xlsr2_300 m.pt
[10][Online]. Available: https://github.com/usnistgov/F4DE

TABLE II
TERM WEIGHTED VALUE COMPARISON BETWEEN THE BASELINE AND THE PROPOSED SYSTEM

Language System	Feature	Pretrain+sp +M=8	Assamese Dev	Assamese Eval	Bengali Dev	Bengali Eval	Pashto Dev	Pashto Eval	Turkish Dev	Turkish Eval	Zulu Dev	Zulu Eval	Average Dev	Average Eval
BeKWS	BNF [11]	✗	17.3	17.9	18.4	17.0	13.5	16.3	29.2	21.6	21.4	22.5	20.0	19.1
JOSTER	BNF	✗	**22.5**	**23.5**	**24.8**	**23.1**	**14.9**	**18.5**	**35.3**	**26.8**	**25.9**	**25.7**	**24.7**	**23.5**
BeKWS	XLS-R	✗	26.4	25.1	29.8	27.4	22.4	26.3	41.2	34.4	31.3	28.9	30.2	28.4
JOSTER	XLS-R	✗	**30.2**	**30.5**	**34.2**	**32.7**	**25.9**	**30.4**	**46.6**	**39.2**	**39.0**	**35.8**	**35.2**	**33.7**
BeKWS	XLS-R	✓	34.0	34.0	35.1	34.2	29.9	33.4	46.0	42.0	39.8	36.2	37.0	36.0
JOSTER	XLS-R	✓	**37.9**	**37.6**	**40.9**	**38.7**	**31.5**	**35.2**	**48.6**	**43.8**	**44.4**	**42.2**	**40.7**	**39.5**

Dev set results are MTWVs while eval set results are ATWVs. "Pretrain+sp+M=8" refers to systems with multilingual pretraining, speed perturbation and increased number of negative training utterances.

units per direction per layer, and a 400-dimensional output projection layer whose outputs are summed along the sequence dimension to obtain the vectoral query representation.

The shared document encoder for JOSTER has 6 bidirectional long short-term memory (BLSTM) layers with 512-dimensional output per direction per layer, followed by a 400-dimensional output layer. We apply dropout of 0.4 between successive BLSTM layers, and down-sample by a factor of 2 after the fourth BLSTM layer. By default (other than in Section IV-E), we do not use any speech-only document encoder between the feature extractor and the shared encoder. This results in document encodings with frame durations of 40 ms for XLS-R features and 20 ms for BNF. The text-only document encoder comprises a 32-dimensional embedding layer, followed by a BLSTM layer with 512-dimensional output per unit per direction, and an affine projection layer to match the input dimension of the shared encoder.

For the baseline (BeKWS), we ensure that the configuration and number of parameters are comparable to the JOSTER configuration above. We use the same query encoder configuration as JOSTER above. We use the configuration of JOSTER's shared document encoder as the document encoder for BeKWS.

For the text-document representation, we set the masking probability to $\pi = 0.3$ and the duration to $\rho = 2$. We obtain these values by tuning to maximize average MTWV on Pashto, Turkish and Zulu dev sets, and apply them without tuning on Assamese and Bengali. For the training loss function, following [11], we set the positive weight to $\lambda = 5$, the tolerance parameter to $\phi = 0.7$ and the number of training utterances per query to $M = 4$.

B. Performance Comparison to BeKWS

In this section, we compare the performance of JOSTER to BeKWS across languages and feature kinds. Table II shows the TWV for each of the five test languages. For the baseline (BeKWS), we note that replacing BNF as used in [11] with XLS-R features yields significant improvements across languages—on average, +10.2 MTWV on the dev sets and +9.3 ATWV on the eval sets. Furthermore, by pretraining the document encoder multilingually for KWS, using speed perturbation and increasing M from 4 to 8 (in (7)), the baseline performance is increased by an additional +6.8 dev MTWV and +7.6 eval ATWV on average across languages, showing that BeKWS with XLS-R features can be improved with multilingual KWS pretraining despite the XLS-R features being already multilingual. This tracks a similar finding about multilingual BNF in [11].

We find that JOSTER *invariably* improves the TWV by considerable margins compared to BeKWS in each setting (BNF, XLS-R, XLS-R + multilingual pretraining). For BNF, the improvements across languages average +4.7 for dev set MTWV and +4.4 for eval set ATWV. When using XLS-R features, the respective improvements increase slightly to +5 and +5.3. When finetuning the multilingually pretrained model with XLS-R features, we get average improvements of +3.7 and +3.5 by using JOSTER instead of BeKWS. Note that we use the same multilingual model—which is trained without unpaired text—to initialize the document encoders for both BeKWS and JOSTER.

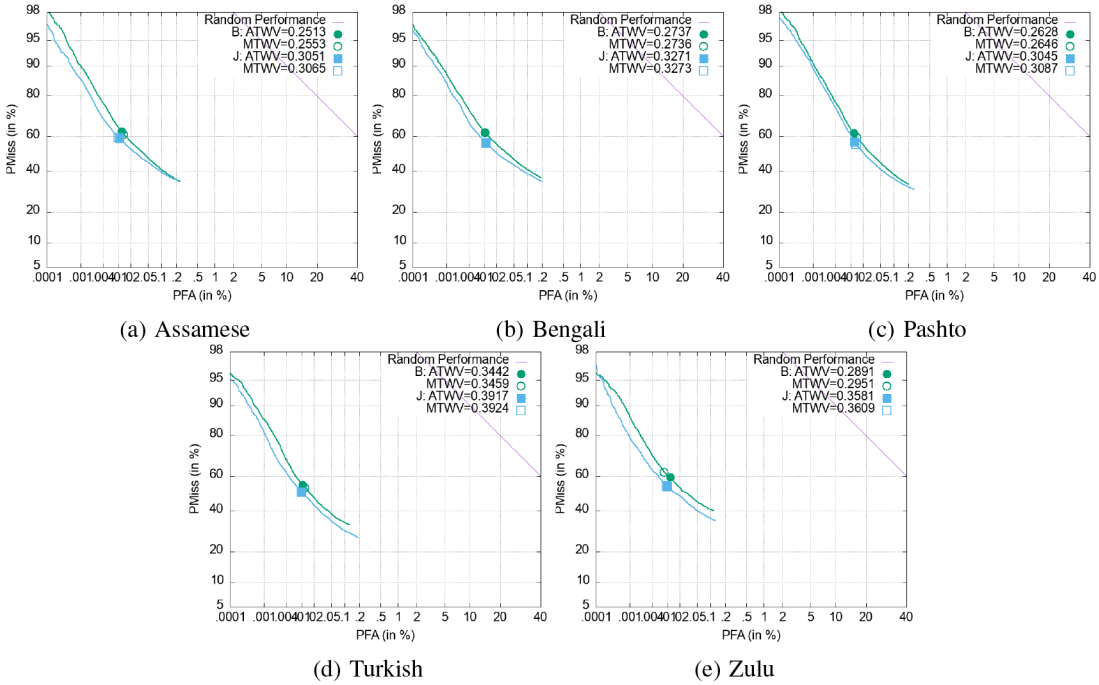
The DET plots in Fig. 3 provide an even more comprehensive picture of the performance difference. In each test language, JOSTER outperforms BeKWS across virtually all operating points of the plots; i.e., at any given recall rate, JOSTER incurs fewer false alarms than BeKWS, further strengthening the significance of the superiority of JOSTER.

C. Text Pre-Processing

As described in Section III-B, when computing the representation of written documents, we first mask with probability $\pi = 0.3$ and repeat each token $\rho = 2$ times. In this section, we quantify the significance of these choices on retrieval performance.

Fig. 4 shows the MTWV as π is varied with ρ fixed to 2. We find that even without masking (at $\pi = 0$), JOSTER already outperforms BeKWS by +3.5 MTWV. This runs counter to our original intuition that without masking, retrieval from written sentences would be too trivial to aid learning. Nevertheless, setting π to 0.15 further improves MTWV by an average of +1.9. Increasing π further starts to worsen performance. We note that although MTWV varies with the masking rate, only at extreme values ($\pi > 0.9$) does it get worse than the baseline, indicating that the joint training is robust across a large range of π. We surmise that having the text input is crucial, and the masking acts as extra regularization in the vein of dropout.

Fig. 5 shows the performance of JOSTER as ρ is varied with π fixed to 0.3. JOSTER outperforms the baseline across all the settings of ρ we tried. Although the average MTWV at $\rho = 2$ is better than the MTWV at $\rho = 1$, by 1.4, the latter may still be preferred as the computational cost of the text document pipeline increases linearly with ρ. Finally, we consider a more involved



(a) Assamese (b) Bengali (c) Pashto

(d) Turkish (e) Zulu

Fig. 3. DET plots showing the evolution of misses and false alarms on the evaluation sets for BeKWS (B) and JOSTER (J).

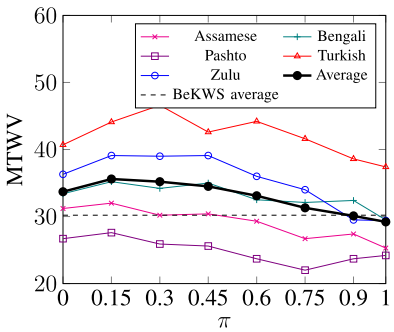


Fig. 4. MTWV on the development sets as the masking rate of text documents is varied.

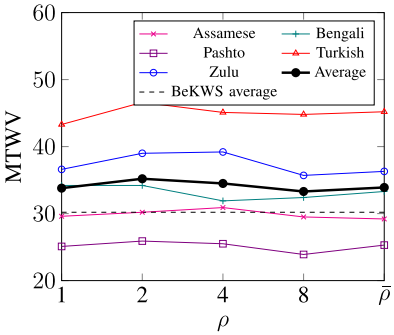


Fig. 5. MTWV on the development sets as the duration of each letter in text documents is varied.

duration model (denoted $\bar{\rho}$ in the figure), where we set ρ for each letter to be its average duration—estimated by forced-alignment with graphemic HMMs trained for each language. We find that this added complexity yields no TWV improvements. In fact, it generally degrades performance compared to fixed duration with $1 \leq \rho \leq 4$.

Overall, we note that although both parameters can change the performance of the system, the variance is low enough that JOSTER still outperforms BeKWS over large ranges of either parameter.

D. Number of Negative Utterances Per Training Step

In this section, we measure the impact of the number of negative examples in each training step. Instead of fixing $M = 4$ for both paired and unpaired batches, we vary them in turn:

- $M(\text{audio})$: We set $M = 4$ for unpaired batches and vary it between 2, 4 and 8 for paired batches.
- $M(\text{text})$: We set $M = 4$ for paired batches and vary it between 2, 4 and 8 for unpaired batches.

Fig. 6 shows the impact as of these variations. In both cases, we find that increasing M increases the MTWV, with $M(\text{audio})$ having higher impact compared to $M(\text{text})$. This however comes at the cost of increased compute and memory cost for each training step. Note that in all our experiments, negative training utterances are sampled randomly. We hypothesize that better sampling of negatives could result in better training efficiency or even better search accuracy, but we leave investigation of any such sampling strategies to future work.

E. Number of Shared Layers

So far, we have fed the speech features directly into the shared encoder, i.e., there are no trainable speech-only encoder parameters. In this section, we reduce the number of shared layers. As we reduce the number of shared layers, we increase

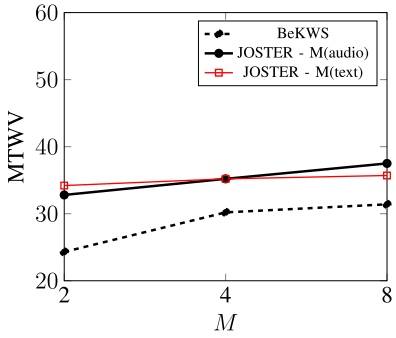


Fig. 6. Average MTWV on the development sets as the number of negative utterances per training step is varied.

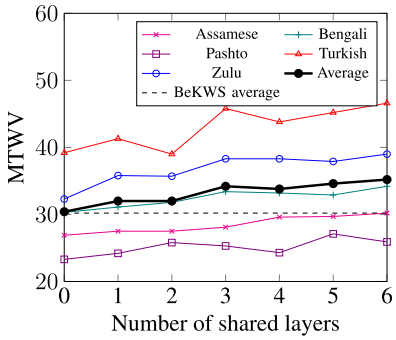


Fig. 7. MTWV on the development sets as the number of layers shared between the two training tasks is varied.

the number of speech-specific layers (including transferring any dropout or down-sampling components) so that the architecture and number of parameters used for the spoken document pipeline ($|\boldsymbol{\Delta}| + |\boldsymbol{\Delta}_{\text{audio}}|$) do not change. For instance, when we remove two LSTM layers from the shared encoder, we use a two layer LSTM network with the same configuration as the speech-only encoder. We keep the text-only encoder configuration constant throughout.

Fig. 7 shows that the MTWV generally improves as more layers are shared. It is particularly noteworthy that when no layers are shared ($\boldsymbol{\Delta} = \varnothing$) by the two modalities' document encoders (with the query encoder shared as always), the MTWV is almost identical to the baseline. This indicates that the performance improvements result from using the unpaired text to improve the (acoustic) representations learned by the document encoder rather than simply having more text data for training the query encoder.

F. Size of Unpaired Text

In this section, we measure the impact as we change the amount of unpaired text used for the auxiliary task and report the results in Fig. 8. First, we compare using the FLP text as has been done so far to using the LLP text, i.e., using the transcripts of the paired data as the "unpaired" text. The LLP text performs significantly worse than using the FLP text and, in three of the five languages, worse even than BeKWS.

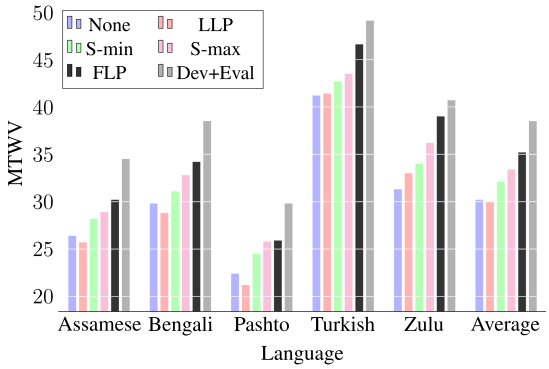


Fig. 8. MTWV on the development sets as the size and composition of unpaired text is varied. None refers to the baseline with no unpaired text, LLP refers to using the LLP text for joint training, S-min denotes the worst of three randomly selected LLP-sized texts while S-max denotes the best of the three, FLP denotes using the entire FLP text for training.

Next, to test how much of this degradation is due to data size and how much of it results from using the same text, we create three random subsets of the FLP text (with the LLP text excluded) each with the same number of sentences as the LLP text. We report the MTWV of the best (S-max) and worst (S-min) performing of these splits for each language. We observe that even the best split performs much worse than the full FLP indicating the size of the augmentation text matters. However, we also observe that even the worst random split outperforms the LLP text, indicating that textual diversity is also crucial. Finally, we report a topline (Dev+Eval) where we use the text from the transcriptions of the Dev and Eval sets as the unpaired text for training, and find that, unsurprisingly, it outperforms using even the larger FLP text. While it is unrealistic to assume that the transcription of the test set can be obtained beforehand, this shows that further improvements can be obtained if it can be somewhat anticipated.

G. In-Vocabulary and Out-of-Vocabulary Queries

In this section, we quantify how much improvement we get on various queries depending on whether or not they exist in the unpaired text.

Fig. 9 shows the ATWV difference between JOSTER and BeKWS across languages for queries that are:

1) *OO:* Out of vocabulary of both the KWS training data and the unpaired FLP text
2) *OI:* Out of vocabulary with respect to the KWS training data but in the FLP text vocabulary
3) *II:* In vocabulary with respect to both KWS training data and the FLP text.

Note that for multi-word queries, out-of-vocabulary means at least one of the query words is out-of-vocabulary.

The worst average improvements over BeKWS (+0.84 on average ATWV) are achieved for OO queries (which form a minute proportion of all queries as shown in the OOV-F column of Table I), with performance even degrading for two of the five languages. For OI and II queries, we get consistent significant improvements (+5.7 and +5.3 average ATWV respective improvements).

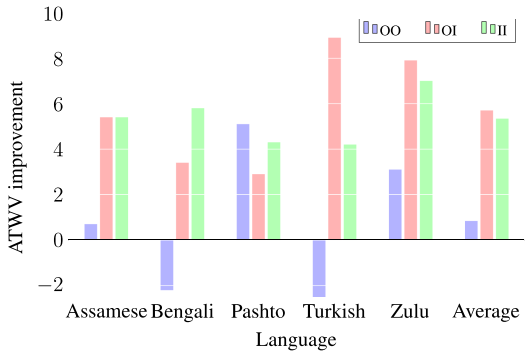


Fig. 9. ATWV differential between the proposed system and the baseline on different subsets of the eval sets' queries. OO denotes queries which are OOV of both the LLP (paired) and FLP (unpaired) training corpora, OI denotes queries which are out of the LLP vocabulary but in the FLP vocabulary and II denotes queries which are in both vocabularies.

Overall, we infer that JOSTER improves the document encoder's representation of words which are in the augmentation text regardless of whether or not they actually exist in spoken form in the paired data.

H. Performance in Mismatched Domain Setting

So far, we have trained and tested exclusively with Babel data. In this section, we experiment with Turkish data from various domains with various configurations of paired and unpaired data. The objective for doing so is twofold:

1) To what extent is the matching domain necessary? In other words, can we improve the TWV in one domain by using text from a different domain?
2) Is text-only domain adaptation possible? Given paired training data in one domain and a test set in another domain, how much can we improve the test set performance using unpaired text from the target domain?

To answer these questions, we conduct experiments on two Turkish language datasets. In addition to the Turkish Babel dataset used in previous experiments, we use Turkish Broadcast News (BNTR) [52] for KWS training and testing.

To match the training data size of the Babel LLP corpus, we use a 10-hour subset of BNTR from the VOA channel[11] for training. This training set has a vocabulary size of 16464. We select two 10-hour subsets from the remaining BNTR data as dev and eval sets. Since the BNTR dataset has no official keyword lists, we randomly select 1500 queries composed of equal proportions of unigrams, bigrams and trigrams for each of the dev and eval sets with OOV rates of 11.7% and 6.5% respectively. We experiment with three text corpora for unpaired training: Babel FLP text, BNTR—unpaired text from the Broadcast News dataset totalling around 180 hours and text from Turkish Wikipedia. The first two allow us to measure the impact of using text from the test domains, while Wikipedia stands as a control corpus.

Table III shows the results of training with various configurations of paired and unpaired data on the different test sets. First, we note that the BNTR results are generally better than the Babel ones, which is to be expected as the latter contains conversational speech from a telephone channel, while the former contains news recordings of professional newscasters.

TABLE III
TWV ON THE TURKISH BABEL AND BROADCAST NEWS DATASETS AS THE PAIRED AND UNPAIRED TRAINING DATA ARE VARIED

System	Paired speech	Unpaired text	BNTR Dev	BNTR Eval	Babel Dev	Babel Eval
BeKWS	Babel	-	55.8	56.1	41.2	34.4
JOSTER	Babel	Babel	64.1	64.1	46.6	39.2
JOSTER	Babel	Wikipedia	68.5	67.6	41.2	37.5
JOSTER	Babel	BNTR	74.6	74.1	46.5	40.8
BeKWS	BNTR	-	84.8	86.1	24.2	20.5
JOSTER	BNTR	Babel	86.5	87.8	29.2	27.4
JOSTER	BNTR	Wikipedia	87.6	88.6	26.1	25.7
JOSTER	BNTR	BNTR	89.0	89.8	28.3	25.6

For each test set, we get improvements by using JOSTER regardless of the unpaired text used for joint training. However, we get the largest improvements when we use text from the test domain to augment training. For instance, in the cross-domain setting where we train with the paired Babel data and test on BNTR, JOSTER with the Babel FLP text improves the dev MTWV and eval ATWV by +8.3 and +8.0 respectively compared to BeKWS. Augmenting with Wikipedia text results in further +4.4 and +3.5 dev and eval improvements compared to using the Babel FLP text. Finally, using BNTR (target domain) unpaired text provides further improvements of +6.1 and +6.5 compared to Wikipedia. This final result cuts the gap to a topline of using BNTR data for training by 65% and 60% on the Dev and Eval sets respectively.

In the converse cross-domain setting (training with BNTR paired data and testing on Babel), we observe a similar trend where JOSTER using Wikipedia improves on the performance of BeKWS, with further performance gains obtained from using BNTR text, and the best performance resulting from using Babel text. We note, however, that the performance improvements are not as dramatic in this case—likely due to the difficulty of transferring the BNTR-trained model to the difficult acoustic conditions of the Babel data.

Finally, when training and testing within the same domain, we observe that JOSTER generally improves the TWV compared to BeKWS even with unpaired text from other domains. This holds even for BNTR which already has a high baseline performance.

Overall, these results add an extra dimension to the results so far, showing that the proposed method performs well, not just across languages—as shown in previous sections—but also across domains within the same language. Furthermore, they suggest that training JOSTER with unpaired text from a domain most improves search performance on test sets in that domain, providing a good alternative when domain-specific data is limited.

I. Comparison With TTS-Based Text Augmentation

In this section, we compare our proposed method with TTS-based unpaired text integration, where we use an off-the-shelf

[11] [Online]. Available: https://catalog.ldc.upenn.edu/LDC2012S06

TABLE IV
TWV ON THE LIBRI-LIGHT CORPUS

System	Unpaired text	Clean Dev	Clean Test	Other Dev	Other Test
BeKWS	-	73.2	72.7	62.2	62.3
JOSTER	Wikipedia	79.2	79.8	67.8	69.5
JOSTER	Librispeech-100	82.6	83.0	71.7	72.7
TTS	Librispeech-100	85.0	84.9	67.8	67.8

TTS model to synthesize speech for the unpaired text and train with the resulting data. Here, we experiment with English language corpora due the difficulty of obtaining high-quality open-source TTS systems for other languages. Specifically, we use the 10-hour Libri-light corpus [53] as the paired KWS training data, and test on the standard Librispeech test splits [54] with around 1300 randomly generated queries for each test split. We use the Coqui xTTS system [55] for speech synthesis. We use the 100-hour Librispeech training set as the unpaired data for JOSTER and TTS. For JOSTER, we also consider training with Wikipedia text.

Table IV shows the results of these experiments. JOSTER, even with Wikipedia text, improves across all dev and test sets, and further, although the best performance is achieved when the in-domain Librispeech-100 text is used. Similarly, using TTS for data augmentation significantly improves KWS compared to BeKWS.

Compared to JOSTER, we note that TTS performs better on the "clean" test sets and performs worse on the more acoustically-challenging "other" sets. This highlights a difference between the two approaches. JOSTER, being text-based, is more channel-agnostic and is more influenced by linguistic similarities between the unpaired text and the target. TTS, on the other hand, is also influenced by channel match between the output of TTS (which is typically clean speech by design) and the test audio. Although, the impact of TTS augmentation on KWS could plausibly be improved by augmentation with artificially generated noise, reverberations or room impulse responses, an in-depth exploration of TTS-based augmentation is out of the scope of this paper. Moreover, these add extra complications which JOSTER does not have.

V. CONCLUSION

In this paper, we propose JOSTER, a method for integrating linguistic context into end-to-end KWS by jointly training a KWS system with an auxiliary text retrieval objective on unpaired text. Furthermore, we conduct experiments comparing the proposed method to a baseline KWS system without the auxiliary objective, and conduct analyses to better understand how the proposed method affects the baseline KWS system. Our experiments show the following:

- The proposed method significantly improves the baseline end-to-end KWS system over several languages and feature types. Moreover, other approaches for improving the baseline such as multilingual pretraining and speed perturbation can also be applied on top of the proposed method to yield further improvements.
- Despite being trained with text, the proposed method improves document (speech) representations rather than query (text) representations of phrases in the auxiliary text. When such phrases are searched, the performance improves regardless of whether the phrase also occurs in the paired training data. On the other hand, the performance on query phrases which are not in the auxiliary text does not improve—and sometimes degrades.
- The proposed approach improves performance even when the auxiliary text is from a different domain than the target test set. However, the best performance is generally achieved when the text domain matches the test set and the proposed approach shows promise as a way to do text-only domain adaptation.

A promising avenue for future work is to extend this approach to other spoken retrieval tasks such as hotword spotting and spoken question-answering for which available paired text-to-text data dwarfs paired speech-to-text data. Another direction is to combine it with semi-supervised training methods so as to be able leverage not just unpaired text but also unpaired speech. Finally, like other E2E-KWS systems, ours relies on inner-product based search in vector spaces, and could therefore benefit from approximate inner-product search methods such as hashing [56] or vector quantization [57], [58] which allow building fast vector indexes with sub-linear memory cost capable of handling up to trillions of documents [59] to make it competitive from a deployment standpoint.

REFERENCES

[1] M. Saraç lar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL 2004: Main Proc.*, 2004, vol. 51, pp. 129–136.
[2] K. Ng and V. W. Zue, "Subword-based approaches for spoken document retrieval," *Speech Commun.*, vol. 32, no. 3, pp. 157–186, 2000.
[3] I. Szöke, M. Fapšo, L. Burget, and J Cernocky, "Hybrid word-subword decoding for spoken term detection," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 42–48.
[4] D. Can and M. Saraçlar, "Lattice indexing for spoken term detection," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 19, no. 8, pp. 2338–2347, Nov. 2011.
[5] C. Chelba, T. J. Hazen, and M. Saraçlar, "Retrieval and browsing of spoken content," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 39–49, May 2008.
[6] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end ASR-free keyword search from speech," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1351–1359, Dec. 2017.
[7] B. Yusuf, A. Gok, B. Gundogdu, and M. Saraclar, "End-to-end open vocabulary keyword search," in *Proc. Interspeech*, 2021, pp. 4388–4392.
[8] J. Švec, L. Šmídl, J. V. Psutka, and A. Pražák, "Spoken term detection and relevance score estimation using dot-product of pronunciation embeddings," in *Proc. Interspeech*, 2021, pp. 4398–4402.
[9] 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.*, 2014, pp. 7844–7848.
[10] C. Chelba, J. Silva, and A. Acero, "Soft indexing of speech content for search in spoken documents," *Comput. Speech Lang.*, vol. 21, no. 3, pp. 458–478, 2007.
[11] B. Yusuf, J. Černocký, and M. Saraçlar, "End-to-end open vocabulary keyword search with multilingual neural representations," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 31, pp. 3070–3080, 2023.
[12] P. Wang, T. N. Sainath, and R. J. Weiss, "Multitask training with text data for end-to-end speech recognition," in *Proc. Interspeech*, 2021, pp. 2566–2570.
[13] B. Yusuf, A. Gandhe, and A. Sokolov, "USTED: Improving ASR with a unified speech and text encoder-decoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 8297–8301.

[14] S. Thomas, B. Kingsbury, G. Saon, and H.-K. J. Kuo, "Integrating text inputs for training and adapting RNN transducer ASR models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2022, pp. 8127–8131.
[15] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 17, no. 8, pp. 1457–1470, Nov. 2009.
[16] C. Liu, A. Jansen, G. Chen, K. Kintzley, J. Trmal, and S. Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *Proc. Interspeech*, 2014, pp. 2789–2793.
[17] B. Gündoğdu, B. Yusuf, and M. Saraçlar, "Joint learning of distance metric and query model for posteriorgram-based keyword search," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1318–1328, Dec. 2017.
[18] B. Gundogdu, B. Yusuf, and M. Saraclar, "Generative RNNs for OOV keyword search," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 124–128, Jan. 2019.
[19] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2009, pp. 421–426.
[20] T. S. Fuchs, Y. Segal, and J. Keshet, "CNN-based spoken term detection and localization without dynamic programming," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2021, pp. 6853–6857.
[21] Z. Zhao and W.-Q. Zhang, "End-to-end keyword search based on attention and energy scorer for low resource languages," in *Proc. Interspeech*, 2020, pp. 2587–2591.
[22] J. Švec, J. Lehečka, and L. Šmídl, "Deep LSTM spoken term detection using Wav2Vec 2.0 recognizer," in *Proc. Interspeech*, 2022, pp. 1886–1890.
[23] D. Can et al., "Web derived pronunciations for spoken term detection," in *Proc. 32nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 83–90.
[24] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in lowresource languages," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2013, pp. 8560–8564.
[25] A. Gandhe, L. Qin, F. Metze, A. Rudnicky, I. Lane, and M. Eck, "Using web text to improve keyword spotting in speech," in *IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 428–433.
[26] G. Mendels et al., "Improving speech recognition and keyword search for low resource languages using web data," in *Proc. Interspeech*, 2015, pp. 829–833.
[27] S. Khurana et al., "A convolutional deep markov model for unsupervised speech representation learning," in *Proc. Interspeech*, 2020, pp. 3790–3794.
[28] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 12449–12460.
[29] A. T. Liu, S.-W. Li, and H.-y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 29, pp. 2351–2366, 2021.
[30] A. Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. Interspeech*, 2022, pp. 2278–2282.
[31] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 267–272.
[32] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2020, pp. 7084–7088.
[33] S. Khurana, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised domain adaptation for speech recognition via uncertainty driven self-training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2021, pp. 6553–6557.
[34] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2020, pp. 7069–7073.
[35] G. Wang et al., "Improving speech recognition using consistent predictions on synthesized speech," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2020, pp. 7029–7033.
[36] M. K. Baskar, L. Burget, S. Watanabe, R. F. Astudillo, and J. Černocký, "EAT: Enhanced ASR-TTS for self-supervised speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2021, pp. 6753–6757.
[37] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end ASR," in *Proc. Interspeech*, 2018, pp. 2394–2398.
[38] M. Wiesner, A. Renduchintala, S. Watanabe, C. Liu, N. Dehak, and S. Khudanpur, "Pretraining by backtranslation for end-to-end ASR in low-resource settings," in *Proc. Interspeech*, 2019, pp. 4375–4379.
[39] Z. Chen et al., "MAESTRO: Matched speech text representations through modality matching," in *Proc. Interspeech*, 2022, pp. 4093–4097.
[40] S. Thomas, H.-K. J. Kuo, B. Kingsbury, and G. Saon, "Towards reducing the need for speech training data to build spoken language understanding systems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2022, pp. 7932–7936.
[41] Y. Tang, J. Pino, X. Li, C. Wang, and D. Genzel, "Improving speech translation by understanding and learning from the auxiliary text translation task," in *Proc. Assoc. Comput. Linguistics-Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4252–4261.
[42] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2017, pp. 5280–5284.
[43] G.-X. Shi, W.-Q. Zhang, G.-B. Wang, J. Zhao, S.-Z. Chai, and Z.-Y. Zhao, "Timestamp-aligning and keyword-biasing end-to-end ASR front-end for a KWS system," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–14, 2021.
[44] R. Yang, G. Cheng, H. Miao, T. Li, P. Zhang, and Y. Yan, "Keyword search using attention-based end-to-end ASR and frame-synchronous phoneme alignments," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 29, pp. 3202–3215, 2021.
[45] R. Huang, M. Wiesner, L. P. Garcia-Perera, D. Povey, J. Trmal, and S. Khudanpur, "Building keyword search system from end-to-end ASR systems," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2023, pp. 1–5.
[46] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015, pp. 3214–3218.
[47] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2012, pp. 336–341.
[48] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The TAO of ATWV: Probing the mysteries of keyword search performance," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 192–197.
[49] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. ACM SIGIR Workshop Searching Spontaneous Conversational Speech*, 2007, pp. 51–57.
[50] "Open KWS14 keyword search evaluation plan," 2013. Accessed: Sep. 2023. [Online]. Available: http://www.nist.gov/itl/iad/mig/upload/KWS14-evalplan-v11.pdf
[51] D. R. Miller et al., "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007, pp. 314–317.
[52] E. Arisoy, D. Can, S. Parlak, H. Sak, and M. Saraçlar, "Turkish broadcast news transcription and retrieval," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 17, no. 5, pp. 874–883, Jul. 2009.
[53] J. Kahn et al., "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., 2020, pp. 7669–7673. [Online]. Available: https://github.com/facebookresearch/libri-light
[54] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, 2015, pp. 5206–5210.
[55] G. Eren and Coqui TTS Team, "Coqui TTS," Jan. 2021. [Online]. Available: https://github.com/coqui-ai/TTS
[56] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, pp. 401–406.
[57] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[58] R. Guo et al., "Accelerating large-scale inference with anisotropic vector quantization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3887–3896.
[59] S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 2206–2240.