Using Pronunciation-Based Morphological Subword Units to Improve OOV Handling in Keyword Search

Yanzhang He, Peter Baumann, Hao Fang, Brian Hutchinson, *Member, IEEE*,
Aaron Jaech, Mari Ostendorf, *Fellow, IEEE*, Eric Fosler-Lussier, *Senior Member, IEEE*, and Janet Pierrehumbert

***Abstract*—Out-of-vocabulary (OOV) keywords present a challenge for keyword search (KWS) systems especially in the low-resource setting. Previous research has centered around approaches that use a variety of subword units to recover OOV words. This paper systematically investigates morphology-based subword modeling approaches on seven low-resource languages. We show that using morphological subword units (morphs) in speech recognition decoding is substantially better than expanding word-decoded lattices into subword units including phones, syllables and morphs. As alternatives to grapheme-based morphs, we apply unsupervised morphology learning to sequences of phonemes, graphones, and syllables. Using one of these phone-based morphs is almost always better than using the grapheme-based morphs, but the particular choice varies with the language. By combining the different methods, a substantial gain is obtained over the best single case for all languages, especially for OOV performance.**

***Index Terms*—Graphones, keyword search, morphological analysis, out-of-vocabulary (OOV) words, speech recognition, subword units.**

I. Introduction

VOCABULARY growth is an important issue for automatic speech recognition, resulting in the twin problems of sparse language model training data and out-of-vocabulary (OOV) words, i.e., words that appear in the test data but are not seen in the training set and thus not represented in the recognizer vocabulary. OOV words are particularly a problem for highly inflective and agglutinative languages, but they can pose challenges for any language in the low-resource setting.

There are three types of applications that tend to have somewhat different approaches to handling OOVs, though all typically involve the use of sub-lexical or subword items in the recognizer vocabulary. For open vocabulary word transcription, subword items are chosen and represented in such a way that orthographic forms can be recovered from the sequence of recognized subwords. In human-computer interaction and voice search, subwords are leveraged to facilitate detection of OOVs and initiate a subdialog for paraphrasing or learning the new word. In keyword search (KWS) or spoken term detection, subwords are used to handle search terms that are OOV. In open vocabulary recognition and keyword search settings, the use of subwords can also help address the data sparsity problem in language model training. In this work, we focus on mitigating OOVs in keyword search, using methods informed by work on open vocabulary recognition. In particular, the subwords being explored are morphology-based units, extending our previously proposed work [1] by introducing alternatives to grapheme-based morphs and experimentation with seven languages.

A variety of methods have been used for deriving subwords, which can be broadly classed as being based on phones or phone n-grams, graphones, syllables, and morphologically based units (possibly including bundles of morphemes) that we will refer to as "morphs." Graphones (orthography coupled with its corresponding phone sequence) [2] and morphs are particularly well suited to open vocabulary recognition. While some work has based the vocabulary entirely on morphs (see [3], [4], [5] and references therein), other studies obtain better results using a combination of morphs and words in Arabic [6] and German [7], [8]. However, a mixed word and syllable vocabulary outperformed a mixed word and morph vocabulary for Polish [9]. A mixed word and graphone vocabulary has also been explored for English [10]. Morphs have the potential advantage of introducing more powerful constraints in language modeling, and several studies have investigated novel language model structures that take advantage of morphological features in a variety of languages [3], [5], [7], [9], [11], [12], [13], [14]. While these studies motivate our use of morphs in this work, only standard n-grams are used here since our focus will be primarily on the keyword search strategies that take advantage of a mixed word and morph or morph-only vocabulary.

Manuscript received January 23, 2015; revised June 20, 2015; accepted October 11, 2015. Date of publication October 30, 2015; date of current version November 17, 2015. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory under Contract W911NF-12-C-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Murat Saraclar.

Y. He was with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA. He is now with Google, Mountain View, CA 94043 USA (e-mail: hey@cse.ohio-state.edu).

P. Baumann is with the Department of Linguistics, Northwestern University, Evanston, IL 60208 USA (e-mail: peter.baumann@u.northwestern.edu).

H. Fang, A. Jaech, and M. Ostendorf are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: hfang@uw.edu; ajaech@uw.edu; mo@ee.washington.edu).

B. Hutchinson is with the Department of Computer Science, Western Washington University, Bellingham, WA 98225 USA (e-mail: brian.hutchinson@wwu.edu).

E. Fosler-Lussier is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: fosler@cse.ohio-state.edu).

J. Pierrehumbert is with the Oxford e-Research Centre, University of Oxford, Oxford OX1 3QG, U.K. (e-mail: janet.pierrehumbert@oerc.ox.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2496222

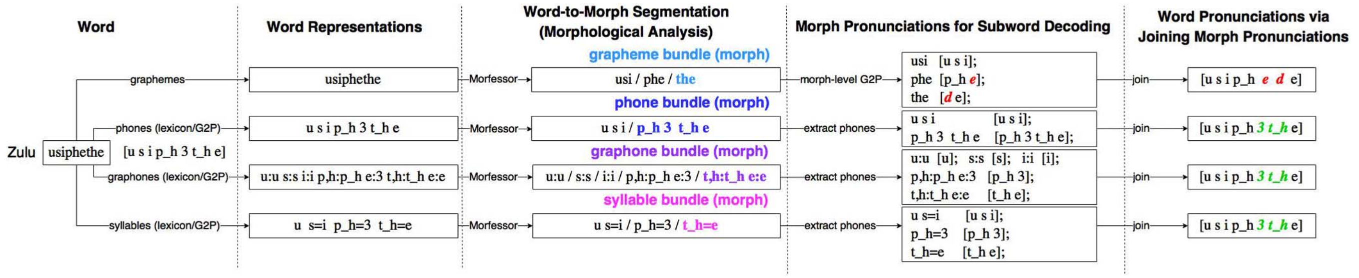

Fig. 1. Illustration of alternative morphological subword units for a Zulu OOV word. "bundle" = pseudo-morph = grouping of graphemes/phones/graphones/syllables. ':' connects graphemes and phonemes as a graphone. '=' connects phonemes as a syllable. '/' separates morphs. Pronunciations are placed inside '[]'. The phones in red font indicate the errors due to the loss of word context in pronunciation prediction for grapheme bundles. The phones in green font are the corresponding correct ones in other bundles.

In keyword search, a standard approach for handling OOVs is to transform a word lattice into a phone lattice when searching for keywords [15], which can be augmented by phone confusions [15], [16], [17], [18]. Directly indexing the output of phone recognition tends to lead to much worse results [15], but in [19], [20], it is shown that decoding with phone n-gram units outperforms the word lattice transformation approach for OOV terms when a flexible segmentation is used to incorporate different order n-grams. Decoding with longer subword units has also been shown to be effective, including pruned phone n-grams [21], [22], phone or character multigrams [23], [24], [25], graphones [26], syllables [27], [28] and morphs. Morphs as subword units are particularly well suited to morphologically rich languages, which have been investigated in [29], [30], [31] and our previous work [1]. In contrast to subword-based approaches, OOV terms can also be searched from word lattices for in-vocabulary (IV) proxies that are phonetically close [18], [32], [33]. In this work, we focus on investigating alternative subword decoding techniques to handle OOVs better. We have not taken advantage of fuzzy search or proxy-based search on the subword decoding systems for possible further improvement due to the higher computation and implementation costs.

Unlike other subword units (e.g., phones), some of the morphs tend to carry meaning and have reasonable average length, leading to a balance between confusability and OOV coverage. Morphs can be derived in an unsupervised fashion from the training corpus for any language using Morfessor [34], which is beneficial especially in the low-resource setting where, for example, syllable annotations are not available.[1]

The majority of previous studies in the literature which use morphs for ASR and KWS typically derive morphs from *grapheme* sequences. However, when morphology learning is unaware of pronunciations, segmentations may occur mid-phoneme, e.g. the Zulu word "ithayima" was segmented by Morfessor as "it/hayi/ma," which incorrectly divides "t" and "h" into two separate morphemes and leads incorrect pronunciations. Even when the segmentation is correct, the lack of context can lead to morph pronunciation errors or confusability. Few studies (if any) have investigated the interaction between morphology and pronunciations that affects ASR and KWS. In this paper we attempt to study whether word *phoneme* sequence (pronunciation) information helps morphology learning for KWS, alone or in combination with graphemes. To answer that, we implement two different pronunciation extraction approaches for grapheme-based morphs and introduce methods for integrating phone information into morphology learning by grouping units that have some phonetic basis (specifically phonemes, graphones or syllables) into a morph instead of grouping grapheme units. To make this more concrete with an example that we later expand on in Fig. 1, we contrast morphology learning for the word "usiphethe" with sequences comprised of:

graphemes (u s i p h e t h e)
phones (u s i p_h 3 t_h e)
graphones (u:u s:s i:i p,h:p_h e:3 t,h:t_h e:e)
syllables (u s=I p_h=3 t_h=e).

In general, we find a benefit from having the morph pronunciations more tightly coupled to the word pronunciations, either via the pronunciation extraction process or morphology learning. Using Morfessor to derive phone-based morphs also tends to result in a more compact subword vocabulary compared to previous studied phone n-grams [19], [20], due to the minimum-description-length principle, which forms the basis of morphology learning.

In our previous pilot study [1], we have shown the effectiveness of using morphs in mixed word and subword decoding for Turkish KWS, and confirmed results from open vocabulary recognition [3] that automatically-derived morphs identified via unsupervised learning using Morfessor can achieve similar performance of morphs identified by a rule-based system designed for Turkish. In this work, we continue the work with unsupervised morphology with the following novel extensions. First, we do a thorough examination of traditional unsupervised grapheme-based morphs with a comparison on 7 low-resource languages that have different "richness" of morphology. We investigate morph and language characteristics that affect KWS performance across languages, and analyze the importance of morph pronunciation and morph length among other factors. Second, we introduce the use of unsupervised morphology learning applied to *phonetic units,* including phonemes, graphones and syllables as alternative solutions to grapheme-based morphs. These alternative morphs differ from the graphone-based morphs introduced in [8], which aligned grapheme-based morphs to graphones, in that here we apply the Morfessor algorithm directly to the phone, graphone

[1] Morfessor's approach is language-independent though it is generally most effective for languages with a concatenative morphology.

or syllable sequence.[2] By asking the learned morphological segmentation to account for the phonetic sequence instead of or in combination with the grapheme sequence, these variants achieve significantly better performance on average across languages for both IV terms and OOV terms than grapheme-based morphs due to more effective segmentations. Third, we show that the combination of the output of both types of systems significantly improves KWS performance especially for OOV terms, due to the diversity of phonetic-unit-based morphs and grapheme-based morphs. Lastly, we explore different decoding and KWS strategies, showing that subword decoding with morphs for OOV terms performs substantially better than lattice/index transformation from word decoding (e.g., [15]). In addition, the combination of the word-only system and the subword-only system performs better than the individual systems or the staging of them (word-based model for IV terms, subword model for OOV terms).

In the sections to follow, we will review KWS in Section II, introduce our methodology in Section III, describe the experiment setup in Section IV, discuss experiment results in Sections V and VI, analyze the results in terms of language differences in Section VII, and conclude in Section VIII.

II. Keyword Search Overview

Keyword search (KWS) is a task to locate all the occurrences of given keyword queries in a corpus of untranscribed speech. We consider in this paper the scenario where the keywords are in text form and are specified after decoding and indexing are done on the speech, which is also known as the task of spoken term detection (STD). KWS has been studied since the 1980's, but it has been a more active area of research in the last decade with a number of competitive evaluations emphasizing conversational speech. In 2006, the U.S. National Institute of Standards and Technology (NIST) initiated an STD Evaluation [35] focused on large-resource languages. Recently, there have been a number of efforts on low resource scenarios and a competitive evaluation associated with the Babel program.[3] Since state-of-the-art LVCSR systems generate far from perfect transcriptions in low-resource languages on conversational speech, lattice outputs of recognition systems are used to improve KWS performance over 1-best outputs, especially in high WER situations [15], [36], [37]. The state-of-the-art KWS systems search keywords from the index of lattices [38], [39], [40] or from the index of confusion networks converted from lattices [41].

A widely used evaluation metric for KWS is the Actual Term Weighted Value (ATWV) [35], introduced in the STD 2006 Evaluation and then becoming the official measure in the Babel program. ATWV is one minus the average loss per term for the actual decision threshold, where loss is a weighted sum of the relative frequency of missed and false detection errors. Since each term contributes equally to the average, the cost of a miss is much more expensive for a rare term than for a term that appears several times [42], which drives the desire for retrieving OOV keyword terms.

Because a global threshold needs to be set across keywords for ATWV calculation, a variety of scoring normalization techniques have been proposed recently [39], [40], [43], [44], [45]. We use the keyword-specific thresholding (KST) approach [43] throughout the paper.

III. Subword-based Decoding and KWS

The general framework for our subword modeling approach is the following. We first segment training words into subwords and then derive their pronunciations. The resulting subwords can be used by themselves in a decoding lexicon, or mixed with the word-based decoding lexicon, in which case the original word-based pronunciations are used for the full words. The segmented training transcripts are utilized to estimate language models for a subword-only vocabulary or a mixed word and subword vocabulary. At test time, the recognition system produces subword-only and/or mixed-unit lattices, from which we search for keywords with their subword and word representations. In KWS, OOV words are decomposed into subword components, and IV words use both the full word and subword decomposition.

The subwords being considered in this paper include variations of morphs and syllables. In the following subsections, we will propose our method for designing alternative types of morphological subword units (Section III-A), and then describe details of unsupervised morphology learning (Section III-B), pronunciation derivation (Section III-C), and language modeling (Section III-D), as well as how morphs are used in keyword search (Section III-E).

A. Subword Alternatives

Morphological decomposition of words is typically done in the orthographic form, which means each word is represented by a sequence of graphemes and split into one or more non-overlapping morphs, so each morph is also represented by a sequence of graphemes. As described next, the decomposition can be automatically derived from the training corpus in an unsupervised fashion through Morfessor, and then used in analyzing new words. The morphology learning algorithm is quite general, and we can provide it other types of sequences for representing words, which can lead to different segmentations. Fig. 1 shows the analysis of a word in Zulu for four different variants that we consider: graphemes, phones, graphones, and syllables. We refer to the morph variants as grapheme/phone/graphone/syllable bundles throughout the rest of the paper, where "bundle" refers to a morphological subword unit (or a pseudo-morph) that groups graphemes, phones, graphones or syllables. (Note that the graphone bundles here differ from the graphone-based morphs used in [7], [8] in that our graphone bundles are learned by applying Morfessor to the graphone representation of words, rather than merging the morphs learned from grapheme-based words with the graphone representation of words.) For comparison, we also use syllables by themselves as subword units without morphological analysis in the experiment sections.

[2] The idea of learning morphs from phonemes was independently and concurrently explored by the Babelon IARPA team, as presented in a July 2014 meeting, but is unpublished and did not consider other phonetically grounded units. Our results were also presented at that meeting.

[3] http://www.iarpa.gov/index.php/research-programs/babel

The derivation of the pronunciation of a morph depends on the symbol sequence it is based on. If the symbol sequence includes phones, then the pronunciation will be more accurate. For the case where the morphs are based on graphemes, there are different options for deriving the pronunciation. The figure illustrates the particular case where the grapheme-based morph pronunciation is derived via grapheme-to-phoneme (G2P) prediction. The example shows that predicting subword pronunciations from subword-level G2P can be error-prone, since the context of the subword within the word is lost, e.g., the grapheme "e" is predicted as /e/ rather than /3/ given only the morph context,[4] while other bundles preserve the word-level pronunciation. Examples where context is important in other languages include: 's' word-finally in English as /s/ vs. /z/; insertion of an inherent vowel /o/ or /O/ when two consonants exist in a row in Assamese or Bengali except when a hoshonto character is used to suppress it; and insertion of an inherent vowel /a/ after a word-final consonant grapheme in Tamil but not a word-internal consonant.

B. Unsupervised Morphology Learning

The subwords were learned in a fully unsupervised manner using the morphological segmentation algorithm *Morfessor Categories-MAP* [34], which has become a standard for unsupervised morphological segmentation [46]. To obtain word-internal segmentations, Morfessor recursively splits words into subwords, and tries to find a lexicon of subwords that is both complete and minimal given the corpus according to a variant of the minimum-description-length principle. The learned subwords are labeled as prefixes (PRE), stems (STM) or suffixes (SUF) using a hidden Markov model to ensure that all words consist of at least one stem with arbitrarily many optional prefixes and suffixes ($(\mathrm{PRE}^* \ \mathrm{STM}\ \mathrm{SUF}^*)^+$ in regular-expression notation).

The degree of segmentation can be manipulated via Morfessor's perplexity threshold parameter b, which controls the likelihood of a given subword being a prefix or suffix in the context of a word. The optimal value of this parameter is usually determined using a labeled development set [34], as its effect strongly depends on the morphological structure of the language and the size of the training set. Since we do not have any labeled data, we used an extensive search over b to find a value that minimizes the percentage of low-frequency subwords (that occur < 5 times) in the lexicon while still providing high coverage of the corpus. Note that this threshold is only used to define a tuning criterion; there are still many low-frequency subwords in the morphological analyses.

For the grapheme-based morphological segmentations, Morfessor was applied to the raw text in standard orthography. For the segmentations based on phonetic subword alternatives, the words were mapped to their phone/graphone/syllable representations and the results of these mappings were fed into Morfessor. The individual phones/graphones/syllables were marked as atoms so that they could not be segmented.

The inventory learned by Morfessor typically includes some single symbol units, but it is not constrained to include all such possible cases and in practice we find that not all are included.

C. Pronunciation Modeling for Subword Units

Pronunciations for subwords are needed both for decoding within the ASR system and for the keyword search system. For OOV words, irrespective of the subword approach, we require a G2P system. We use the lexicon of the training set to train our grapheme-to-phoneme system to predict word-level pronunciations for OOV keywords. We follow a joint multigram approach utilized by the Phonetisaurus G2P toolkit [47]. This system predicts pronunciations based on a multigram alignment between graphemes and phonemes; we train the alignment model using the pronunciation lexicon for in-vocabulary words.

For grapheme-based morphs, we implement two different methods for deriving pronunciations. First, we use the standard G2P model trained on words to predict a single pronunciation for each grapheme morph as would be done for new vocabulary items. A limitation of this approach is the lost context for short morphs, which can lead to poor pronunciations as shown in the figure. As an alternative, we align each IV word's dictionary pronunciation to its morph sequence to extract morph pronunciations, which yields multiple pronunciations for each morph that occurs in different words. In the case of the example in the figure, the "the" morph would then have both [d e] and [t_h e] as alternative pronunciations, among a total of 10 variants, which are all added to the subword decoding lexicon. This approach leads to better coverage of the actual pronunciation, but potentially more confusability. In either case, the grapheme-based segmentations pose a challenge when the segmentation break occurs mid-phoneme.

The phone bundle system uses the trained G2P system on the OOV word to first predict pronunciations as a sequence of phones, and then uses this as the representation to Morfessor for morphological segmentation. An advantage is that pronunciations are immediately readable from the morph identity, but any graphemic clues to the morphological segmentation are lost.

By placing additional constraints on the G2P model, we can also derive graphonic and syllabic representations for input to Morfessor. The graphone-based system uses the G2P model to find the best alignment between graphemes and phonemes; the search is constrained to predict zero or more phonemes for every grapheme (one-to-many alignment), but in post processing graphemes with null pronunciation are combined with the subsequent graphone pair (e.g., "p:- h:p_h" becomes "p,h:p_h" in the graphone example in Fig. 1). This effectively annotates the grapheme-based system with phonetic information. On the other hand, the syllable-based system constrains the G2P system to produce valid syllable structures as pronunciations [27], which are then used as the input representation for Morfessor.

For all but the G2P grapheme-based morphs, we find the morph sequence for an OOV word by first applying G2P to obtain the word-level pronunciation, optionally in graphone or syllable form. Morfessor is used with the grapheme or other symbol sequence to find the all possible segmentations. For the grapheme-based models, each morph in a segmentation is then

[4] In this paper phones are written using the X-SAMPA symbols used in the Babel language packs, which can be found at http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm.

associated with a pronunciation using one of the two options above. Note that because the morph inventory does not include all possible single symbols, there will be some words that cannot be segmented with a particular morph inventory.

D. Subword-Based Vocabulary and Language Model

In our experiments, we consider different variations on the vocabulary and language model (LM), including some using only subword units and some using a mix of subword and word units. All subword units are marked by % which distinguishes them from the word tokens, and different types of subword units, e.g., prefixes, stems, and suffixes, are indicated by the position of the %. For example, the English word “unbreakable” would be segmented to “un%”, “%break%” and “%able” using this notation. We train trigram, 5-gram and 7-gram LMs with SRILM [48] using modified-Kneser-Ney smoothing. To control for model size, we use entropy pruning to reduce the 5-gram and 7-gram LMs to be approximately the same size as the corresponding trigram LM. However, in Section VI-D we show that OOV keyword search performance using 5-gram and 7-gram are comparable in our setting. Therefore, we use trigrams for the remainder of the paper.

Subword-Only LM: In the subword-only scenario, the training data is fully expanded into subwords using the most likely word-to-subword decomposition.[5] Language models are trained on this expanded text, using a vocabulary that consists only of the subword tokens in the expansion of the words observed in the training set.

Mixed-Unit LM: In the mixed-unit scenario, three segmentations of the data are used to train the language models, including the original word segmentation, the fully expanded (subword-only) segmentation, and a mixed word-subword segmentation that has been selectively expanded (details below). In all component models for the mixed-unit LM for a particular subword type, the vocabulary is the set of all words observed in the training set and all subwords in the expanded version of these words. We produce four mixed-unit language models:

1) A “fully decomposed” model trained on the fully expanded data. Note that this model differs from the subword-only LM in that this model includes all words in the vocabulary with a small probability.[6]
2) A “partially decomposed” model trained on the selectively expanded data.
3) A “2-interp” LM that is an equally weighted interpolation[7] of the previous two LMs.
4) A “3-interp” LM that is an equally weighted interpolation of the first two LMs and an LM trained on the original text (words without segmentation).

The partially expanded text used in the “partially decomposed” model is produced by expanding only a subset of the words in the text into their subwords, leaving other words intact. The “expansion set” of words to be decomposed includes all words w that do *not* meet any of the following (tunable) criteria: i) w appears more than θ_1 times; ii) one of w’s subwords would appear fewer than θ_2 times in the expanded text; or iii) one of w’s subwords appears in fewer than θ_3 words. A simple, iterative algorithm is used to find the expansion set satisfying the above criteria using $\theta_2 = \theta_3 = 5$ (to avoid introducing infrequent subword units) and $\theta_1 = 500$ (to ensure that the most frequent words were left intact).

E. Keyword Search

Our keyword search algorithm is based on index lookup: for a word decoding system, we create a word-based index from the lattices, tracking all of the words that occur in the lattice, their start and end times, and their lattice posterior probabilities. We use “lattice-tool” from SRILM [48] to convert lattices to indices, where the time axis is quantized into points with T seconds interval and T is optimized to 0.1 in our experiments. Both start and end times of each hypothesized word are mapped into the closest quantized time points, where same word occurrences with the same times are merged by adding the posteriors. The merged index has the flavor of confusion networks [49] in the sense of summing the posteriors and providing alternative paths that might not be in the lattice for a multiword term. For single-word keywords, we return the list of all keyword occurrences, sorted by their posterior probabilities. For multiword keywords, we retrieve the individual words from the index in the correct order with respect to their start and end times but discard occurrences where the time gap between adjacent words is non-zero with respect to the quantized time points.[8] We approximate the multiword posterior with the minimum of the individual word probabilities as in [40], which we also found slightly better than the product of the posteriors. All the hypotheses of a keyword form a posting list. The detection threshold in the list is determined separately for each keyword using an empirical estimate of each keyword’s term weighted value (TWV) [42]. The probabilities in each keyword’s posting list are normalized using KST-normalization [43] to enable a single, keyword-independent, detection threshold.

For a mixed-unit decoding or subword-only decoding system, we follow the same search and thresholding algorithms, but the index units would be whatever are chosen as the decoding units, namely mixed words and subwords, or only subwords. For the case of mixed-unit decoding, we also augment the index by adding subwords expanded from the decoded words. During search, each word of a keyword is represented by the word itself and a subword sequence if it can be segmented. We consider all

[5] If a word has multiple pronunciations in the provided dictionary, then each pronunciation will have a different decomposition for phone, graphone and syllable bundles. Preliminary experiments suggested that using the decomposition based on the first pronunciation for the purpose of LM training works as well as determining the pronunciation by forced alignment, and simplifies the processing chain.

[6] This model is trained by estimating an n-gram trained on the subwords with the constraint to match marginals to a unigram distribution that is the interpolation of the subword unigrams and a uniform distribution over the full word vocabulary, using a heuristically chosen interpolation weight.

[7] We notice in our preliminary experiments that the interpolation weights have minimal impact on the keyword search performance. As it is expensive to tune the interpolation weights based on the output of the speech recognition system or the keyword search system, we fix the interpolation weights to be uniform among LMs.

[8] Initial versions of our system [1] allowed for a 0.5 second gap between keywords, but in later experiments we found after tuning the allowable gap distance on the development set that no gap reduced the false alarm rate, and thus improved ATWV. In fact, time quantization already effectively allows a gap between adjacent words to recover misses.

possible subword sequences if there are multiple segmentations from different pronunciations for a word. For a single-word keyword, we search from the mixed-unit or subword-only indices for all representations of the word following the index lookup approach described above. For a multiword keyword, its representation for search would be the cross product of all the representations of each component word. Currently each representation for a keyword is equally weighted for simplicity. KWS hyper-parameters (e.g., LM scale and posterior scale) are tuned on development data separately for each system using the Nelder-Mead optimization method [50].

Staging or system combination can be used for utilizing multiple individual KWS systems. For staging/cascade (e.g., [15]), keywords that cannot be found with previous systems are searched using subsequent systems. For example, we can search IV keywords from word-decoded lattices, then search OOV keywords from morph-decoded lattices, and finally search phone lattices for OOV keywords that cannot be covered by the IV morph inventory. In our initial experiments on Turkish using grapheme bundle decoding, staging with phone lattices had minimal impact, likely due to the high OOV coverage using the morphs and the poor performance of the word lattice expansion approach. Therefore, we decided not to do staging for OOVs that cannot be covered. In further experiments, we found that a system combination strategy gave better performance than staging. Specifically, we use an approach that integrates the posting lists from different systems and combines the scores (e.g., [39], [43]). We tuned the posterior and language model scaling factors for the best single system, and applied the same factors to all other systems to ensure the score ranges are compatible. The posting lists from each system are merged by averaging the detection probabilities of overlapping entries. KW-specific detection thresholds are determined using a decision theoretic criterion [40]. Detection probabilities are then normalized as in [43], and a single threshold for all KWs is determined by Maximum Term Weighted Value (MTWV) [35] on the development data. For OOV keywords, we also use this strategy for combining multiple systems based on different types of subword units.

IV. Experiment paradigm

A. Data Description

We evaluate the effectiveness of our proposed methods to handle OOVs in the keyword search task on seven low-resource languages provided by the IARPA Babel Program. We use the conversational telephone speech portion of the 10-hour limited language pack (LimitedLP) training set for each language, which has word-level transcriptions and a pronunciation lexicon. The development set for each language also contains 10 hours of speech with transcriptions. Our evaluation set is the transcribed "eval-part1" set, which has about 15 hours of speech for Tamil, and 5 hours for each of the other languages. The development set is used to tune parameters, which are applied to the evaluation set. We report results on both sets. The official evaluation keyword list for each language is used for all experiments. Table I lists the versions of the language packs and keyword lists for all the data we use.

TABLE I
Babel Data Description for Seven Low-Resource Languages Investigated in this Work

Language	Version	Keyword List
Zulu	IARPA-babel206b-v0.1e	conv-eval.kwlist4
Turkish	IARPA-babel105b-v0.4	conv-eval.kwlist2
Tagalog	IARPA-babel106b-v0.2g	conv-eval.kwlist2
Haitian-Creole	IARPA-babel201b-v0.2b	conv-eval.kwlist4
Assamese	IARPA-babel102b-v0.5a	conv-eval.kwlist4
Bengali	IARPA-babel103b-v0.4b	conv-eval.kwlist4
Tamil	IARPA-babel204b-v1.1b	conv-eval.kwlist5

TABLE II
Vocabulary Size, LM Perplexity (PPL) and Word Error Rates (%) for Dev and Eval Sets with Word Decoding Baseline Systems Using a Trigram LM

		Dev		Eval	
Language	Vocab (k)	PPL	%WER	PPL	%WER
Zulu	13.6	59	70.3	59	72.0
Turkish	10.1	194	65.9	207	65.7
Tagalog	5.5	117	61.6	114	60.5
Haitian-Creole	4.8	130	63.4	126	62.5
Assamese	7.7	61	65.2	64	65.0
Bengali	7.9	66	67.6	73	65.1
Tamil	14.3	250	76.6	235	77.5

TABLE III
Keyword Statistics (IV and OOV) in the Development and Evaluation Sets

Language	Keywords in Dev			Keywords in Eval		
	IV	OOV	%OOV	IV	OOV	%OOV
Zulu	1067	310	22.5	1031	380	26.9
Turkish	1298	387	22.9	1173	452	27.8
Tagalog	1228	510	29.3	1084	651	37.5
Haitian-Creole	1308	124	8.6	1248	287	18.6
Assamese	1185	176	12.9	1349	259	16.1
Bengali	1298	206	13.6	1311	283	17.7
Tamil	1213	267	18.0	1682	499	22.8

B. ASR System

We use the Kaldi toolkit [51] to build a single automatic speech recognition system prior to keyword search. Standard 13-dim PLP features combined with 3-dim Kaldi pitch features [52] are first extracted as input for maximum likelihood GMM-HMM model training. The features are then transformed by linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT). They are further adapted by feature-space maximum likelihood linear regression (fMLLR), which is estimated by speaker adapted training (SAT). The GMM-HMM models are retrained with the resulting features to provide the alignment for subsequent DNN-HMM hybrid system training. A DNN with tanh neurons is trained using the same speaker-adapted features. The details of the DNN training are documented in section 2.2 in [53]. The baseline language model for word decoding is a trigram with modified-Kneser-Ney smoothing and pruning. We follow the default setup to train the acoustic model with position-dependent

TABLE IV
SUBWORD DECODING VS. INDEX EXPANSION FOR OOV KEYWORDS ON THE *DEVELOPMENT* SET. #FA IS THE NUMBER OF FALSE ALARMS

	Decoding Unit	LM	Index Expansion	OOV ATWV	OOV #Hit	OOV #FA	OOV #Miss	%OOV Recall (Lattices)
Zulu	word	word	grapheme bundle	0.037	33	245	546	22.5
		word	syllable	0.081	62	566	517	42.8
		word	phoneme	0.093	68	516	511	60.8
	grapheme bundle	subword-only	-	**0.167**	113	491	466	42.1
Turkish	word	word	grapheme bundle	0.026	47	389	904	14.1
		word	syllable	0.035	44	292	907	13.9
		word	phoneme	0.064	69	424	882	51.7
	grapheme bundle	subword-only	-	**0.119**	133	509	818	29.6

triphones for word-based decoding. We train another acoustic model with position-independent triphones for subword-based decoding so that it can be reused for different types of subword units.[9] The word error rates for the baseline word decoding systems are reported in Table II.

C. KWS Experiments

In the keyword lists, only those keywords that exist in the data set would be counted towards the ATWV score. We list in Table III the number of keywords that exist in either set. The OOV keyword rates range from 8.6% to 37.5%. Note that the term “keyword” in this paper refers to a query term, each of which could be a word or a phrase. A keyword is considered an OOV keyword when it has at least one OOV word in it. To handle OOVs, we decode with the components and search strategy described in Section III.

We present the KWS results in the next two sections. In Section V, we describe a set of experiments using the grapheme bundle morphs for subword decoding and keyword search. In these experiments, we explore different decoding and indexing alternatives and show the benefit of explicit subword decoding for OOVs over the traditional word lattice/index expansion approach. In Section VI, we compare the keyword search performance across subword alternatives and show that adding phonetic information in subword learning improves the system performance over grapheme-based learning. In both sections we analyze language differences. Since these differences imply that there is no single best subword strategy, the best results are obtained by the combination of all subword units.

V. GRAPHEME-BASED MORPH EXPERIMENTS

In this section, we empirically study the performance of the typical grapheme-based morphs to explore the effect of decoding vocabulary, keyword search strategy and pronunciation modeling across languages, in order to identify a good configuration for subsequent experiments with different types of subwords.

A. Subword Decoding vs. Index Expansion

For OOV keywords, we compare two approaches to construct subword indices: either by expanding the word indices from a word-decoded system to subword indices, or by obtaining subword indices directly from a subword-decoded system. The OOV results for three methods of expansion (grapheme bundles, syllables, phones) are compared to grapheme-bundle subword decoding in Table IV for Zulu and Turkish; results for other languages have a similar trend. No subword LMs are used in the expansion. The grapheme bundles for subword-only decoding use the morph-level G2P predicted pronunciations.

In both languages, subword decoding achieves much higher OOV ATWV than all of the index expansion approaches; phoneme expansion works better than syllable expansion, which in turn is better than grapheme bundle expansion. Note that phoneme search is much more time- and memory-consuming than any other methods. We measure the recall rate for OOV keywords in the lattices, i.e. the percentage of OOV keywords that have matches in the lattices (regardless of whether their scores are above the KWS detection threshold). Phoneme expansion has the highest recall as expected, so its posterior scores must be worse than subword decoding due to the lower ATWV. Subword decoding with grapheme bundles not only has similar or higher recall than expanding word graphs with grapheme bundles or syllables, but also has two or three times as many hits as all the index expansion systems. These results suggest that using subword decoding (with subword LMs) is important for recognizing subword sequences reliably for OOVs.

B. Mixed-Unit Decoding vs. Subword-Only Decoding

Table V shows in detail ASR and KWS results of three different decoding vocabularies: words, mixed unit (words and subwords), and subwords only, where the subwords are grapheme bundles. Search level parameters, like lattice beam, are fixed across languages, except language model weight, which is tuned per language after lattices are generated. For the purpose of calculating WER, lattices of the two subword-based systems are first transduced into word-based lattices by composing with a subword-to-word transducer which encodes word segmentations. Note that in this case, the WER for mixed and subword system must be taken with a grain of salt: we would expect WER performance to be worse than the corresponding word-based system because, for example, the subword hypothesis space will not have the advantage of a word-based language model.

The lattice densities for these systems are shown in Table VI; lattices are more compact for subword systems. For the word-based system, the OOV ATWV results are achieved by phoneme search via index expansion. All systems use trigram language

[9] “Position" refers to word position (or subword position for subword decoding). Switching from position-dependent to position-independent triphones modeling for subword decoding has minimal impact on the KWS performance.

TABLE V
VOCABULARY SIZES, WORD ERROR RATES AND OOV/IV/OVERALL ATWV RESULTS FOR WORD-ONLY DECODING VS. MIXED-UNIT DECODING VS. SUBWORD-ONLY DECODING WITH *GRAPHEME BUNDLE* UNITS ON THE *DEVELOPMENT* SETS. RESULTS IN BOLD INDICATE THE BEST PERFORMANCE ACROSS THREE SYSTEMS FOR EACH EVALUATION METRIC IN EACH LANGUAGE RESPECTIVELY

		Decoding Unit		
		word-only	mixed-unit	subword-only
Zulu	Vocab (k)	13.6	17.1	3.5
	%WER	**70.3**	73.5	75.8
	OOV	0.093	**0.175**	0.167
	IV	**0.345**	0.291	0.278
	All	**0.288**	0.265	0.253
Turkish	Vocab (k)	10.1	12.9	2.8
	%WER	**65.9**	69.3	70.2
	OOV	0.064	**0.119**	**0.119**
	IV	**0.302**	0.254	0.237
	All	**0.247**	0.223	0.210
Tagalog	Vocab (k)	5.5	8.4	3.0
	%WER	**61.6**	64.9	70.4
	OOV	0.018	0.113	**0.125**
	IV	**0.404**	0.365	0.265
	All	0.291	0.291	0.224
Haitian-Creole	Vocab (k)	4.8	8.3	3.5
	%WER	**63.4**	66.0	64.8
	OOV	0.044	**0.137**	**0.137**
	IV	**0.352**	0.330	0.320
	All	**0.326**	0.313	0.304
Assamese	Vocab (k)	7.7	8.7	1.1
	%WER	**65.2**	72.0	74.6
	OOV	0.035	0.042	**0.060**
	IV	**0.289**	0.191	0.154
	All	**0.256**	0.172	0.142
Bengali	Vocab (k)	7.9	8.9	1.0
	%WER	**67.6**	75.2	79.1
	OOV	**0.064**	**0.065**	**0.064**
	IV	**0.284**	0.171	0.121
	All	**0.254**	0.157	0.114
Tamil	Vocab (k)	14.3	15.7	1.5
	%WER	**76.6**	80.3	83.1
	OOV	**0.051**	0.042	0.035
	IV	**0.282**	0.173	0.154
	All	**0.237**	0.149	0.132

TABLE VI
RECOGNITION LATTICE DENSITIES (THE AVERAGE NUMBER OF ARCS THAT CROSS A FRAME) FOR WORD-ONLY DECODING VS. MIXED-UNIT DECODING VS. SUBWORD-ONLY DECODING WITH *GRAPHEME BUNDLE* UNITS ON THE *DEVELOPMENT* SETS

	Decoding Unit		
	word-only	mixed-unit	subword-only
Zulu	934	889	799
Turkish	848	831	662
Tagalog	758	906	589
Haitian-Creole	1199	1621	667
Assamese	1493	1021	854
Bengali	1737	1058	865
Tamil	3194	1346	1024

TABLE VII
OOV ATWV AND OOV WORD PRONUNCIATION PHONE ERROR RATE (PER) COMPARISON OF TWO GRAPHEME BUNDLE SUBWORD-ONLY SYSTEMS WITH DICTIONARIES BASED ON DIFFERENT APPROACHES TO EXTRACT MORPH PRONUNCIATIONS. RESULTS ARE REPORTED ON THE *DEVELOPMENT* SET. RESULTS IN BOLD INDICATE BETTER PERFORMANCE BETWEEN THE TWO SYSTEMS FOR EACH EVALUATION METRIC IN EACH LANGUAGE RESPECTIVELY

	OOV ATWV		OOV Pron. %PER	
Grapheme Bundle Pron.	predicted	aligned	predicted	aligned
Zulu	**0.167**	0.144	11.5	**3.0**
Turkish	0.119	**0.126**	7.4	**1.5**
Tagalog	0.125	**0.151**	17.8	**3.0**
Haitian-Creole	0.137	**0.143**	3.2	**2.4**
Assamese	**0.060**	0.057	13.0	**3.4**
Bengali	**0.064**	0.045	10.1	**4.7**
Tamil	0.035	**0.053**	5.2	**0.2**

models. The LM for the mixed-unit system is the 2-way interpolation of the fully and partially segmented training text. The word-based system has stronger LMs for in-vocabulary words than subword-based systems, leading to clearly higher IV ATWV and lower WER. The weaker subword-only language model gives more opportunity for OOV words to be represented in the lattice, so usually achieves the highest ATWV on the OOV terms and beats the phone expansion of the word-based system for OOVs on all languages except Tamil. The mixed-unit decoding tends to do better than the subword-only system on IV words (and even better when using the 3-way interpolation LM that includes the word n-grams), but not as good as the word-based system. The mixed-unit system is rarely better than the subword-only system, and we find that it is better to use system combination than mixed unit decoding to combine the benefits of words and subwords for KWS. We focus on the subword-only systems for the rest of the experiments because of their simplicity and better performance on OOVs.

C. Pronunciation Modeling

Directly predicting pronunciations at the morph level can be error-prone due to the loss of word context, so we also explore morph pronunciation extraction based on alignment of the morph sequence to IV word pronunciations, as described in Section III-C. The two different pronunciation derivations are compared in terms of OOV ATWV using subword-only decoding systems. Results on the development keyword set are reported in Table VII, where "predicted" means the morph pronunciations are predicted at the morph-level using the trained G2P model and "aligned" indicates that the morph pronunciations are generated through word-level grapheme-phoneme alignment. We obtained moderate system improvement in 4 of the 7 languages by using the "aligned" morph pronunciations. A likely reason why the "aligned" pronunciations in some cases degrade performance is because they introduce multiple pronunciations for each morph, which could also increase the confusability during decoding. As illustrated by the example in Section III-C, this is particularly an issue for Zulu. Table VII also shows the OOV word pronunciation phone error rate (PER), comparing the actual error for the "predicted" model and the oracle error for the "aligned" model, where

TABLE VIII
Base Unit Inventory Sizes for Graphemes, Phones, Graphones and Syllables

	#Graphemes	#Phones	#Graphones	#Syllables
Zulu	28	49	368	1107
Turkish	33	44	139	1675
Tagalog	29	41	290	1848
Haitian-Creole	30	33	103	1979
Assamese	63	47	313	1518
Bengali	62	48	340	1769
Tamil	48	33	126	2202

the oracle is obtained by choosing the lowest PER option of the multiple possible morph sequence pronunciations. The oracle PER is an optimistic estimate, but is used based on the assumption that the acoustic model would tend to match the lower PER option in recognition. As expected, the "aligned" morph pronunciation model better matches the true OOV word pronunciations, but the reduction in PER is not indicative of improvement in ATWV because it does not account for the confusability.

VI. Phonetic Morph Experiments

In our second set of experiments, we explore different choices of subword units as alternatives to grapheme bundles. Instead of the original grapheme representation of words, we can represent a word with a sequence of phones, graphones or syllables. With the method described in Section III-A, we use Morfessor on top of these representations to automatically learn phone bundles, graphone bundles and syllable bundles as pseudo-morphs respectively. In addition, we consider syllables by themselves as another alternative subword type.

In the experiments, we mainly use grapheme bundles with word-level aligned pronunciations as the baseline and for combination, but we also show results of grapheme bundles with morph-level predicted pronunciations as reference.

A. Morph Length & OOV Coverage

Table IX lists subword statistics for different languages. The phone inventories for Assamese, Bengali and Tamil are smaller than their grapheme inventories (Table VIII), which consist of Indian scripts. The inventory size contributes to the morph length, since in general unsupervised morph learning based on a bigger base unit inventory tends to generate shorter morphs in order to gain sufficient occurrences in multiple words. As a result, the average morph length (measured by "#phones/morph", Table IX) of their derived phone bundles is larger than that of their grapheme bundles - a grapheme in the three Indian languages typically corresponds to a phoneme. Therefore, phone bundles can effectively solve the over-segmentation problem of grapheme bundles for languages with these characteristics.

Graphone bundles are shorter than both grapheme bundles and phone bundles in the non-Indian languages likely due to the larger graphone inventories. However, for Indian languages graphone bundles are still almost as long as grapheme bundles, suggesting that the phonetic information helps morph learning (more discussion in Section VII).

TABLE IX
Vocabulary Statistics for Different Subword Units. "bd" is Short for "Bundle". Pronunciations for Grapheme Bundle are Generated by Word-Level G2P Alignment. Best Case Unit(s) for OOV ATWV for Each Language is Indicated in Bold

Language	Subword Unit	#Morphs / Word	#Phones / Morph	#Morphs	OOV KW %Coverage
Zulu	grapheme bd.	2.8	4.5	3506	94.8
	phone bd.	2.7	4.3	4438	95.2
	graphone bd.	4.1	3.5	2125	94.8
	syllable bd.	3.3	3.9	2025	91.0
	syllable	3.9	2.9	1107	97.1
Turkish	**grapheme bd.**	2.6	4.0	2823	94.1
	phone bd.	2.3	4.2	3840	86.0
	graphone bd.	3.3	3.7	2380	95.3
	syllable bd.	2.8	3.7	3043	46.8
	syllable	3.2	2.8	1675	57.6
Tagalog	**grapheme bd.**	2.1	4.3	2985	88.8
	phone bd.	2.0	4.3	3685	90.2
	graphone bd.	3.5	2.8	1955	93.5
	syllable bd.	2.6	3.6	2362	81.6
	syllable	2.9	3.0	1848	93.9
Haitian-Creole	grapheme bd.	1.6	4.2	3527	89.5
	phone bd.	1.8	3.8	2650	87.1
	graphone bd.	2.9	2.6	1098	96.8
	syllable bd.	2.0	3.4	2401	54.8
	syllable	2.1	3.1	1979	67.7
Assamese	grapheme bd.	3.2	3.2	1099	95.5
	phone bd.	2.0	4.2	3500	90.9
	graphone bd.	2.9	3.0	2265	89.2
	syllable bd.	2.6	3.4	2272	76.1
	syllable	2.9	2.9	1518	86.9
Bengali	grapheme bd.	3.2	3.1	997	98.1
	phone bd.	1.9	4.0	3963	88.3
	graphone bd.	2.8	3.0	2221	93.7
	syllable bd.	2.4	3.4	2709	68.4
	syllable	2.6	2.9	1769	86.4
Tamil	grapheme bd.	3.5	3.6	1537	99.3
	phone bd.	2.6	4.7	5035	91.8
	graphone bd.	3.3	3.8	2351	93.6
	syllable bd.	3.0	3.9	3993	84.3
	syllable	3.6	3.1	2202	93.3

Syllables as subword units are relatively short, each of which has about 3 phones on average for all languages, and the syllable boundaries need to be annotated by language experts (provided by the language packs). Syllable bundles are derived in such a way that they can make use of the syllable boundaries and are longer than syllables for less confusability.

"OOV KW Coverage" in Table IX reports the percentage of OOV keywords that can be fully represented by IV subwords in the form of their segmentations (which is different than the keyword OOV rate). Typically shorter subword types have higher coverage.

A reason for the generally better coverage of the morphs than syllables is that the Morfessor algorithm explicitly optimizes a minimum description length criterion, which finds an efficient code for characterizing words based on available training data.

TABLE X
NUMBER OF UNIQUE SINGLE PHONES AS GRAPHEME BUNDLES, PHONE BUNDLES AND SYLLABLES, AND NUMBER OF PHONES FOR EACH LANGUAGE

	#phones	#Unique single phones in		
		grapheme bd.	phone bd.	syllable
Zulu	49	27	42	7
Turkish	44	30	34	16
Tagalog	41	24	34	0
Haitian-Creole	33	30	30	12
Assamese	47	33	37	15
Bengali	48	38	42	13
Tamil	33	32	32	12

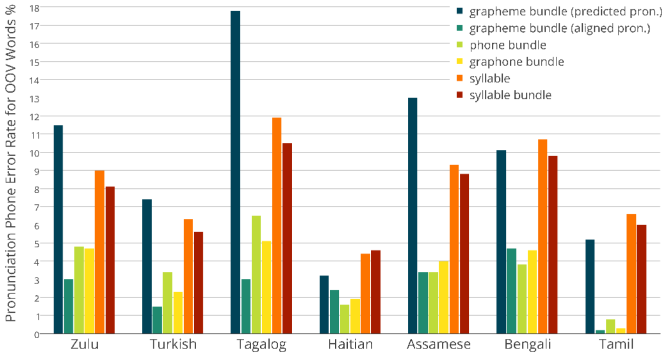

Fig. 2. Phone error rate for development set keyword OOV word pronunciations comprised of subword pronunciations.

Depending on the complexity of syllables in the language, this can be more or less of an advantage. The morphs do contain substantially more single phones than do syllables (Table X), but they do not include all possible phones and the number of single phones is not a good indicator of OOV keyword coverage. In particular, the languages with the fewest single phone syllables (Zulu, Tagalog) have the highest coverage rates for syllables ($> 93\%$). The limited training scenario also plays a role: Morfessor has worse coverage for syllable bundles due to the large symbol set of syllable. In order to enhance the coverage of syllable bundles, we expand the syllable bundle index back to a syllable index before keyword search.

B. Morph Pronunciation

We construct pronunciations of OOV words in keywords by putting together the pronunciations of all component subword units. We measure the phone error rates of such OOV word pronunciations in Fig. 2 and present the differences using different subword types.[10]

Besides the grapheme bundles with aligned pronunciations, phone bundles and graphone bundles have the lowest PER, because they are derived after the word-level G2P is applied to OOV words. The phone error rates of phone and graphone bundles are effectively 1/3–1/2 of that of grapheme bundles with predicted pronunciations. For syllables and syllable bundles, since each predicted syllable is forced to follow a legitimate syllable structure learned during G2P training, the syllable-based pronunciations are less accurate than those based on phone bundles or graphone bundles.

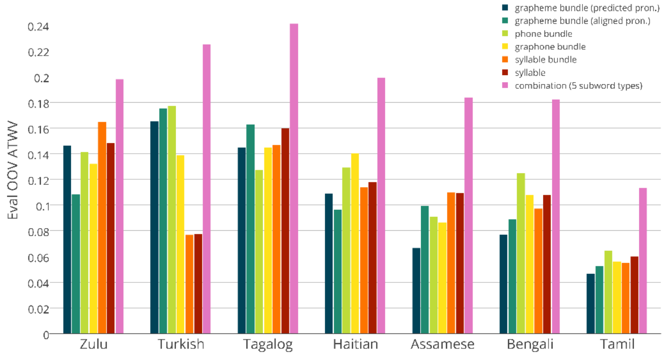

Fig. 3. OOV ATWV on the *evaluation* sets with subword-only decoding using 5 different subword types, and with the combination of the 5 systems.

The IV PER is 1.1–14.2% for grapheme bundles with predicted pronunciations, lower than their OOV PER, but 0 for other subword types since their pronunciations are aligned against the true word pronunciations from the training lexicon.

C. KWS Results

Keyword search results for OOV keywords based on subword-only decoding are presented in Fig. 3 for all 5 types of subword units. Adding phonetic information for morph learning helps in general as the consequence of multiple factors including reduced G2P errors, longer units and increased OOV coverage. The best subword type varies due to characteristics of the languages. For all 3 Indian languages, all of the phonetic subword types perform well due to the improved pronunciations. The poor results for Turkish using syllables and syllable bundles are likely due to their low coverage for OOV keywords (57.6%, Table IX). Grapheme bundles perform reasonably well for languages where they are relatively long. Phone bundles seem to be the best choice in general because they are simple and effective in most languages, and they do not require human annotation of syllables. The combination of all 5 subword units improves OOV ATWV substantially, which indicates their diversity can reduce misses of OOV keywords although at the cost of increased false alarms.

In order to further investigate when the corrected subword pronunciations improve KWS performance, we also compare the IV ATWV results across different subword types in Fig. 4. For the IV words, the 4 non-grapheme-bundle subword types all have perfect pronunciations, and all subword types have perfect coverage. Phone bundles improve IV ATWV over grapheme bundles for all languages except for Haitian-Creole likely due to the already low PER for its grapheme bundles (1.1%). This improvement for Tagalog, Assamese and Bengali is even bigger than that in OOV keywords, partly due to their higher IV PER reduction compared to other languages. The other 3 non-grapheme-bundle subword types are also better than grapheme bundles except for Zulu and Haitian-Creole since apparently morph length is still a confounding factor. Because coverage is not a problem for IV keywords, Turkish syllables and syllable bundles perform similarly to or slightly worse than other bundles. Phone bundles are the best or among the best consistently in all languages. The combination of all subword

[10] If a word has multiple pronunciations provided by the development set lexicon, we compare the predicted pronunciation with all of them and report the lowest PER.

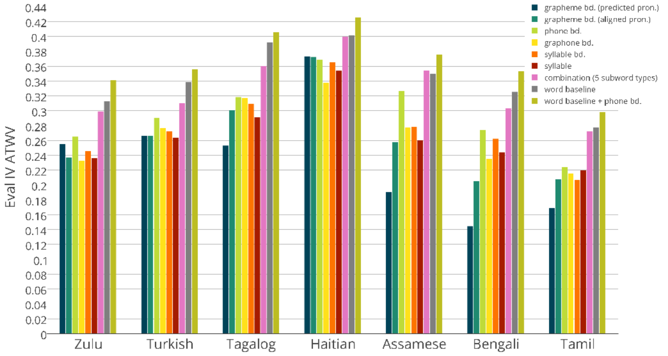

Fig. 4. IV ATWV on the *evaluation* sets with subword-only decoding using 5 different subword types, with the combination of the 5 systems, with the word-only decoding baseline, and with the combination of the word-only decoding baseline and the phone-bundle-only decoding system. "bd" is short for "bundle".

units is less effective for IV keywords than OOV keywords, likely because OOV ATWV is much more sensitive to missed detection penalties according to the definition. However, the word decoding baseline system performs even better than the combination of all 5 subword systems for IV keywords. The best system can be achieved by combining the word baseline with the phone bundle system. Adding other subword systems leads to minimal further improvement.

In most languages for both OOV and IV ATWV, the morphological subword units based on phonetic information perform better than the grapheme bundles with aligned pronunciations and even better than the ones with predicted pronunciations. This suggests that not only do better morph pronunciations help, but also adding pronunciations into morphology learning is useful.

D. Effect of Higher Order LMs

In ASR experiments for morphologically rich and concatenative languages, it has previously been shown, for example in [4], that longer n-grams are beneficial for shorter sub-words. We experimented with long-span n-grams for subword-only LMs, comparing against trigrams for the purpose of OOV keyword search and ASR. The results are given in Table XI. We conducted this set of experiments on Bengali since it has both a relative long subword type (phone bundle, 1.9 morphs per word) and a short subword type (grapheme bundle, 3.2 morphs per word). For fair comparison, the number of n-grams for 5-gram and 7-gram LMs are pruned using entropy pruning to be approximately the same as that for 3-gram LMs. As shown in Table XI, WER correlates with OOV ATWV well when different subword types are compared. For each subword type, the difference of OOV ATWV across LM order is minimal. Higher order LMs seem to perform slightly worse than or just about the same as trigrams in terms of WER. We also tried using 5-gram and 7-gram LMs that were not pruned to match the trigram size, but still did not observe any significant performance improvement. In our case, it appears that neither ASR nor KWS benefit from such higher-order context, perhaps due to the data sparsity caused by our 10-hour training set. Based on these results, we fixed our LM choice to be trigram throughout the paper. Further evaluations on more data and more languages could be future work for interested readers.

TABLE XI
OOV ATWV AND WER ON THE *DEVELOPMENT* SET USING DIFFERENT N-GRAM ORDER SUBWORD-ONLY LMS FOR EACH SUBWORD TYPE FOR BENGALI. PRONUNCIATIONS FOR GRAPHEME BUNDLE ARE GENERATED BY WORD-LEVEL G2P ALIGNMENT. AVERAGE NUMBER OF MORPHS PER WORD FOR EACH SUBWORD TYPE IS ALSO LISTED

Subword Unit	#Morphs	OOV ATWV			WER		
	/ Word	3-gr	5-gr	7-gr	3-gr	5-gr	7-gr
grapheme bd.	3.2	0.045	0.042	0.047	72.5	72.9	73.0
phone bd.	1.9	0.115	0.109	0.107	69.7	69.8	69.8
graphone bd.	2.8	0.094	0.088	0.092	70.3	70.6	70.7
syllable	2.6	0.069	0.075	0.072	73.0	73.1	73.1

TABLE XII
SYSTEM COMBINATIONS FOR IV, OOV AND OVERALL ATWV ON THE *EVALUATION* SETS. SYSTEMS: S0) WORD DECODING BASELINE: WORD SEARCH FOR IV AND PHONE SEARCH FOR OOV (VIA INDEX EXPANSION). S1) IV: S0 + PHONE BUNDLE SYSTEM; OOV: PHONE BUNDLE SYSTEM. S2) IV: S1; OOV: COMBINATION OF SYSTEMS OF 5 SUBWORD TYPES

Language	System	IV	OOV	All
Zulu	S0	0.313	0.079	0.250
	S1	0.341	0.159	0.292
	S2	0.341	0.198	0.303
Turkish	S0	0.339	0.043	0.257
	S1	0.356	0.182	0.307
	S2	0.356	0.225	0.320
Tagalog	S0	0.392	0.035	0.258
	S1	0.406	0.135	0.305
	S2	0.406	0.241	0.344
Haitian-Creole	S0	0.401	0.026	0.331
	S1	0.425	0.122	0.369
	S2	0.425	0.199	0.383
Assamese	S0	0.350	0.050	0.302
	S1	0.375	0.091	0.330
	S2	0.375	0.184	0.344
Bengali	S0	0.325	0.103	0.286
	S1	0.353	0.114	0.311
	S2	0.353	0.182	0.323
Tamil	S0	0.278	0.044	0.224
	S1	0.298	0.061	0.244
	S2	0.298	0.114	0.256

E. System Combination

The final system combination results are shown in Table XII. The baseline is a word decoding system where OOV search is handled by expanding word indices into phones. Using the phone bundle subword-only decoding system for OOV and combining it with the baseline for IV already provides good gains over the baseline, which leads to significant improvement in the overall ATWV with up to 0.05 absolute difference. These results suggest that using morphology-based subword units for decoding, especially those learned with phonetic information, is effective to handle the data sparsity issue in the low-resource setting for keyword search. This combination is efficient because only one subword decoding is needed in addition to the word decoding. If resources allow, combining all 5 types of subword units for the OOV portion can achieve further improvement in overall ATWV with up to 0.086 absolute difference compared to the word baseline, but with the overhead of 5 decoding systems.

VII. Analysis of Language Differences

System performance for the different subword units varies across languages, so it is of interest to examine language differences. Zulu, Turkish and Tamil are agglutinative languages with rich morphology. They have the largest vocabularies (#IV words), leading to low trigram hit rates, high OOV rates and high WER for recognition (Table II). Tagalog also has a rich morphology, but it is less agglutinative and has non-concatenative morphological features, such as infixation and reduplication. Conversational Zulu and Tagalog have a high ratio of code switching with English, which partially causes the high G2P phone error rates. Haitian-Creole has very limited productive morphology, lacking any form of inflectional marking, but since it is based on French, its words still reflect French derivational morphology. Like many other Indo-European languages, Assamese and Bengali are fusional languages, i.e. their rich morphological systems involve non-concatenative morpheme combinations and changes in the form of the word stem triggered by different morphemes.

Table IX statistics show that the grapheme bundles for the three Indian languages (Assamese, Bengali and Tamil) are relatively short (#phones per morph) so that they have the highest OOV keyword coverage but at the cost of high acoustic and lexical confusability. The large grapheme sets could be a cause of this over-segmentation issue, which contributes to the worse KWS performance on the Indian languages. Since shorter units lead to high acoustic and lexical confusability, the grapheme-based systems perform considerably worse than the phone-based systems for the three Indian languages. This difference for IV words is even more pronounced. We fit a simple linear regression on the vocabulary statistics in Table II and VIII to predict the morph length for each language and found that the log of vocabulary size and the base unit set size (#graphemes) are most correlated with the length of the derived grapheme bundles. For the four non-Indian languages, despite the fact that they are in different places on the spectrum of morphological “richness” - Zulu and Turkish are highly agglutinative, Tagalog has a rich, but partially non-concatenative morphology, and Haitian-Creole has only very limited morphology - the morph-based subword decoding approaches works reasonably well on all of them.

One factor that appears to affect morph length resulting from unsupervised learning is the base unit inventory size: in general, a bigger inventory tends to produce shorter morphs in order to obtain sufficient occurrences in multiple words. The three Indian languages have bigger grapheme sets than the other languages, which cause the over-segmentation issue for their grapheme bundles; their smaller phone sets lead to phone bundles being longer than grapheme bundles while maintaining good coverage. Another factor is phonetic regularity. When we used a graphone set for the Indian languages, despite it being large, it leads to fewer morphs per word for graphone bundles than grapheme bundles, likely because it makes the grapheme set more specific. According to our one-to-many alignment generation process, graphones are really graphemes annotated with phonetic information - including this information allows Morphessor to find more coherent, longer chunks. Switching from graphemes to phones, graphones or syllables, the inherent default vowel of consonants (see Section III-A) becomes explicitly visible.

Besides morph length, there are other factors leading to system performance differences. Syllable coverage is an issue for Turkish OOV keywords. Pronunciation PER also has an impact, but it is difficult to assess its role - PER reduction correlates with IV ATWV improvement reasonably well but not with OOVs since it interacts with morph length, OOV coverage and other factors. In theory, Morfessor is expected to be more effective on languages with concatenative morphology, but it does not appear to be a deciding factor experimentally - Zulu and Turkish perform well on OOV keywords while Tamil does not. Besides concatenative morphology, the language-inherent acoustic confusability certainly plays a role as well. Tamil is the hardest even in the case of word decoding (Table II), since their words are confusable in terms of pronunciation. All of these factors interact to affect both ASR and KWS performance across languages.

VIII. Conclusion

In this paper, we have systematically investigated the usage of morphological subword units in KWS for handling the OOV issue in the low-resource setting. We evaluate their effectiveness in 7 languages that have different degrees of morphology, ranging from highly agglutinative languages like Zulu to languages with limited productive morphology like Haitian-Creole. We show that the morphology-based subword approach is effective in all languages but requires careful choices of system components. First of all, subword decoding is better than subword expansion from word decoding for handling OOVs; it provides better subword posterior estimates. Furthermore, pronunciations have an effect on morphological decomposition and hence ASR and KWS performance. For grapheme-based morphs, extracting morph pronunciations based on alignments to whole word pronunciations gives lower PER than predicting pronunciations for these morphs in isolation, and for most languages results in improved KWS. However, better results can be obtained by using subword definitions that learn from pronunciations, either by applying morphology learning to phone or graphone sequences or by using syllables. These novel morphological subword units have not only reduced pronunciation errors, but have also learned morphology from phonetic regularity, which improved KWS performance in both IV and OOV keywords. In addition, the combination of multiple types of morphs is able to obtain substantial improvement over individual systems especially on OOV performance at the cost of multiple decodings.

We find that grapheme bundles on Indian languages do not work well out of the box since the words are over-segmented, which is possibly due to the specifics of the Indian scripts and their relatively large grapheme sets combined with the small amount of text available for morphology learning. The proposed phone bundles have effectively solved both issues and have better pronunciations, which turn out to be longer and perform substantially better in KWS than grapheme bundles.

As for all experimental work in speech recognition, the improvements obtained here need to be interpreted in the context of the particular speech recognition technology used. In this work, we have not taken advantage of fuzzy or proxy-based keyword search techniques due to the higher computation and implementation costs, but that may provide an alternative approach for achieving performance gains. The rapid advances in neural network-based systems may impact the findings. However, algorithms for unsupervised morphology are also advancing with the increasing interest in low resource languages, which could increase impact and/or benefit from the generalization to more phonetically-based analysis.

Although we focus on the application of keyword search, approaches proposed here could be adapted to other domains like open vocabulary recognition and OOV detection. For example, in open vocabulary recognition, word transcriptions can be derived directly from the graphemes associated with graphone morphs.

Future work may benefit from more sophisticated morphological feature-based approaches in language modeling which would provide better models of long-span subword dependencies, and better rescoring of putative subword hits in keyword posting lists. Phonetically close IV morph sequences to a keyword segmentation can be generated based on confusions, which can be used in a fuzzy search strategy for detecting keywords that are not covered or reducing misses for other keywords. Considering code switching or morphology learning for G2P might be useful to further reduce morph pronunciation errors for languages like Zulu and Tagalog. In addition, a principled framework that allows for tuning the morphological analysis pipeline based on ASR and KWS performance directly would potentially be helpful.

Acknowledgment

The authors would like to thank Steven Wegmann, Adam Janin and Hang Su for their help on the ASR system and the KWS baseline system. We are also grateful to anonymous reviewers for their comments and suggestions.

References

[1] Y. He, B. Hutchinson, P. Baumann, M. Ostendorf, E. Fosler-Lussier, and J. Pierrehumbert, "Subword-based modeling for handling OOV words in keyword spotting," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 7864–7868.

[2] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2005, pp. 726–729.

[3] H. Sak, M. Saraçlar, and T. Güngör, "Morpholexical and discriminative language models for Turkish automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 8, pp. 2341–2351, Oct. 2012.

[4] T. Hirsimäki, J. Pylkkönen, and M. Kurimo, "Importance of high-order n-gram models in morph-based speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 724–732, May 2009.

[5] E. Arisoy, M. Saraçlar, B. Roark, and I. Shafran, "Discriminative language modeling with linguistic and statistically derived features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 540–550, 2012.

[6] A. El-Desoky, C. Gollan, D. Rybach, R. Schlüter, and H. Ney, "Investigating the use of morphological decomposition and diacritization for improving Arabic LVCSR," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2009, pp. 2679–2682.

[7] A. E.-D. Mousa, M. A. B. Shaik, R. Schlüter, and H. Ney, "Sub-lexical language models for German LVCSR," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2010, pp. 171–176.

[8] M. A. B. Shaik, A. E.-D. Mousa, R. Schlüter, and H. Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary German LVCSR," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2011, pp. 1441–1444.

[9] M. A. B. Shaik, A. E.-D. Mousa, R. Schlüter, and H. Ney, "Using morpheme and syllable based sub-words for polish LVCSR," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2011, pp. 4680–4683.

[10] M. A. B. Shaik, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Hierarchical hybrid language models for open vocabulary continuous speech recognition using WFST," in *Proc. Workshop Statist. Percept. Audit.*, 2012, pp. 46–51.

[11] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational Arabic speech recognition," *Comput. Speech Lang.*, vol. 20, no. 4, pp. 589–608, 2006.

[12] M. A. B. Shaik, R. Schlüter, and H. Ney, "Investigations on the use of morpheme level features in language models for Arabic LVCSR," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, pp. 5021–5024.

[13] M. A. B. Shaik, H.-K. J. Kuo, L. Mangu, and H. Soltau, "Morpheme-based feature-rich language models using deep neural networks for LVCSR of Egyptian Arabic," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8435–8439.

[14] E. Whittaker and P. Woodland, "Particle-based language modelling," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2000, pp. 170–173.

[15] M. Saraçlar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. Conf. North Amer. Chap. Assoc. Comput. Linguist.: Human Lang. Technol. (NAACL-HLT)*, 2004, pp. 129–136.

[16] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraçlar, "Effect of pronounciations on OOV queries in spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2009, pp. 3957–3960.

[17] I. Bulyko, O. Kimball, M.-H. Siu, J. Herrero, and D. Blum, "Detection of unseen words in conversational Mandarin," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012, pp. 5181–5184.

[18] M. Saraçlar, 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. Autom. Speech Recogn. Understand. Workshop (ASRU)*, 2013, pp. 464–469.

[19] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Subword speech recognition for detection of unseen words," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2012, pp. 2446–2449.

[20] D. Karakos and R. Schwartz, "Subword and phonetic search for detecting out-of-vocabulary keywords," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2014, pp. 2469–2473.

[21] O. Siohan and M. Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2005, pp. 53–56.

[22] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2009, pp. 404–409.

[23] K. Ng and V. W. Zue, "Subword-based approaches for spoken document retrieval," *Speech Commun.*, vol. 32, no. 3, pp. 157–186, 2000.

[24] I. Szoke, L. Burget, J. Cernocky, and M. Fapso, "Sub-word modeling of out of vocabulary words in spoken term detection," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2008, pp. 273–276.

[25] W. Hartmann, V.-B. Le, A. Messaoudi, L. Lamel, and J.-L. Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2014, pp. 2764–2768.

[26] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006spoken term detection system," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2007, pp. 2393–2396.

[27] H. Su, J. Hieronymus, Y. He, E. Fosler-Lussier, and S. Wegmann, "Syllable based keyword search: Transducing syllable lattices to word lattices," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2014, pp. 489–494.

[28] T. Mertens and D. Schneider, "Efficient subword lattice retrieval for German spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2009, pp. 4885–4888.

[29] V. T. Turunen and M. Kurimo, "Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval," *ACM Trans. Speech Lang. Process.*, vol. 8, no. 1, pp. 1–25, 2011.

[30] S. Parlak and M. Saraçlar, "Performance analysis and improvement of Turkish broadcast news retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 731–741, Mar. 2012.

[31] K. Narasimhan, D. Karakos, R. Schwartz, S. Tsakalidis, and R. Barzilay, "Morphological segmentation for keyword spotting," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP)*, 2014, pp. 880–885.

[32] B. Logan and J.-M. Van Thong, "Confusion-based query expansion for OOV words in spoken document retrieval," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2002, pp. 1997–2000.

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

[34] M. Creutz and K. Lagus, "Inducing the morphological lexicon of a natural language from unannotated text," in *Proc. Int. Interdisciplinary Conf. Adaptive Knowl. Represent. Reasoning*, 2005, vol. 1, no. 106–113, pp. 51–59.

[35] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006spoken term detection evaluation," in *Proc. ACM SIGIR Workshop Searching Spontaneous Conversat.*, 2007, pp. 51–55.

[36] F. Seide, P. Yu, C. Ma, and E. Chang, "Vocabulary-independent search in spontaneous speech," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2004, pp. 253–256.

[37] 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.

[38] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 615–622.

[39] J. Mamou, J. Cui, X. Cui, M. J. F. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schluter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 8272–8276.

[40] D. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. Lowe, R. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2007, pp. 314–317.

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

[42] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, "The TAO of ATWV: Probing the mysteries of keyword search performance," in *Proc. Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2013, pp. 192–197.

[43] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V.-B. Le, "Score normalization and system combination for improved keyword spotting," in *Proc. Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2013, pp. 210–215.

[44] O. Vinyals and S. Wegmann, "Chasing the metric: Smoothing learning algorithms for keyword detection," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 3301–3305.

[45] V. Soto, L. Mangu, A. Rosenberg, and J. Hirschberg, "A comparison of multiple methods for rescoring keyword search lists for low resource languages," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2014, pp. 2464–2468.

[46] M. Kurimo, S. Virpioja, V. T. Turunen, G. W. Blackwood, and W. Byrne, "Overview and results of Morpho challenge 2009," in *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, 2010, pp. 578–597.

[47] J. Novak, N. Minematsu, and K. Hirose, "WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding," in *Proc. Int. Workshop Finite State Meth. Nat. Lang. Process.*, 2012, pp. 45–49.

[48] A. Stolcke, "SRILM—an extensible language modeling toolkit," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, 2002.

[49] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Comput. Speech Lang.*, vol. 14, no. 4, pp. 373–400, 2000.

[50] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optimizat.*, vol. 9, no. 1, pp. 112–147, 1998.

[51] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. Autom. Speech Recognit. Understand. Workshop (ASRU)*, 2011, pp. 1–4.

[52] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 2494–2498.

[53] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 215–219.

Yanzhang He, photograph and biography not available at the time of publication.

Peter Baumann, photograph and biography not available at the time of publication.

Hao Fang, photograph and biography not available at the time of publication.

Brian Hutchinson, photograph and biography not available at the time of publication.

Aaron Jaech, photograph and biography not available at the time of publication.

Mari Ostendorf, photograph and biography not available at the time of publication.

Eric Fosler-Lussier, photograph and biography not available at the time of publication.

Janet Pierrehumbert, photograph and biography not available at the time of publication.