MULTILINGUAL REPRESENTATIONS FOR LOW RESOURCE SPEECH RECOGNITION AND KEYWORD SEARCH

Jia Cui[1*], Brian Kingsbury[1], Bhuvana Ramabhadran[1], Abhinav Sethy[1], Kartik Audhkhasi[1], Xiaodong Cui[1], Ellen Kislal[1], Lidia Mangu[1], Markus Nussbaum−Thom[1], Michael Picheny[1], Zoltán Tüske[2], Pavel Golik[2], Ralf Schlüter[2], Hermann Ney[2], Mark J. F. Gales[3], Kate M. Knill[3], Anton Ragni[3], Haipeng Wang[3], Phil Woodland[3]

[1] IBM Watson, 1101 Kitchawan Rd, Yorktown Heights, NY, 10598, U.S.A.

[2] Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany

[3] Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK

ABSTRACT

This paper examines the impact of multilingual (ML) acoustic representations on Automatic Speech Recognition (ASR) and keyword search (KWS) for low resource languages in the context of the OpenKWS15 evaluation of the IARPA Babel program. The task is to develop Swahili ASR and KWS systems within two weeks using as little as 3 hours of transcribed data. Multilingual acoustic representations proved to be crucial for building these systems under strict time constraints. The paper discusses several key insights on how these representations are derived and used. First, we present a data sampling strategy that can speed up the training of multilingual representations without appreciable loss in ASR performance. Second, we show that fusion of diverse multilingual representations developed at different LORELEI sites yields substantial ASR and KWS gains. Speaker adaptation and data augmentation of these representations improves both ASR and KWS performance (up to 8.7% relative). Third, incorporating un-transcribed data through semi-supervised learning, improves WER and KWS performance. Finally, we show that these multilingual representations significantly improve ASR and KWS performance (relative 9% for WER and 5% for MTWV) even when forty hours of transcribed audio in the target language is available. Multilingual representations significantly contributed to the LORELEI KWS systems winning the OpenKWS15 evaluation.

Index Terms— Multilingual Representation, Hierarchical Deep Neural Network, Keyword Search, BABEL

1. INTRODUCTION

Multilingual (ML) models have been shown to outperform unilingual models for ASR in low resource languages [1–8]. Recently, ML models have also exhibited great advantages in keyword search (KWS) tasks such as Babel [9–12] and the Spoken Web Search Task held as part of MediaEval Benchmark [13]. This paper focusses on the impact of different ML representations on IBM's speech recognition and keyword search systems used in the Babel Optional Period 2 (OP2) surprise language evaluation. These ML features were independently developed at RWTH Aachen (RWTH) [11, 14, 15], Cambridge University (CUED) [16, 17] and IBM.

Multilingual ASR has been investigated over the last two decades. The approaches can be broadly classified into two main categories: one, focused on generating universal language-independent lexicons and the second, focused on language independent acoustic representations. With the recent success of Deep Neural Networks (DNNs), and their ability to generalize and learn useful acoustic representations of languages, focus has shifted to using DNN-derived multilingual representations. The approach that we take in this work, belongs to the second category. The advantage of using ML features for a time-limited evaluation like BABEL is that the ML features themselves can be trained in advance of the evaluation period. However, given the large amount of data (approximately 1000 hours spanning $10+$ languages) used for ML training, this process can be very time consuming.

Several methods have been proposed in the literature to speed up training of neural networks. While an exhaustive review of such methods is beyond the scope of this paper, we mention a few relevant techniques here. Optimization techniques that parallelize training across multiple machines have also been explored for DNN training [18–22]. However, these methods involve significant communication costs. In order to reduce these data communication costs [23] proposed a 1-bit quantization of gradients with nearly no loss in accuracy, while [24] proposed a combined hardware/software solution. An alternative approach uses data sampling to speed up training. [25] presents a methodology for using varying sample sizes in batch optimization methods for large scale machine learning problems. The authors propose a criterion for dynamic sample selection in the evaluation of the function and gradient based on variance estimates obtained during the computation of a batch gradient.

In deriving multilingual representations, there have been studies focussed on carefully identifying a subset of language(s) closest to the target language [26, 27], with subsequent use of data from this subset only for training the network. Not only do these networks offer performance improvements, they train faster by virtue of use of less data. In this paper, we present a data sampling strategy, that allows the network to see the training data across all languages in stages. We show that this can shorten the training time to one third of the original time with less than a 1% relative loss in speech recognition performance.

Next, we present the impact of various input features used to derive multilingual representations. The IBM ML representations are derived from the log-Mel filterbank spectrum. In contrast, the input features used by our partner sites, are alternate features, such as gamma-tone features, RASTA PLP [28,29] etc. We demonstrate that the performance of all these multilingual representations are compa-

*Corresponding author, Email: jiacui@us.ibm.com

rable regardless of the input space.

We also revisit techniques which have proven to be helpful for speech recognition and keyword search in previous years of the BABEL program. These include, re-alignment during training, semi-supervised learning (SSL), and data augmentation. In this work, the above techniques are re-examined under the context of ML representations.

The rest of the paper is organized as follows. Section 2 decribes the IBM ML framework, analyzing the proposed data sampling strategy with respect to accuracy and training speed ups. Section 3 briefly introduces the Babel OP2 task and its three conditions, as well as IBM ASR and KWS systems. Section 4 presents the recognition performance of diverse ML representations for the two low resource conditions studied. A comparison of unilingual models, SSL models and speaker-adapted models is also presented. Section 5 presents our preliminary results on adapting the ML features to the target language. Section 6 demonstrates the impact of the ML representations on keyword search performance. Finally, we analyze the use of ML representations in KWS for both, low-resource (3 hours) and medium resource conditions (40 hours). The paper concludes with key messages in Section 7.

2. A SIMPLE HIERARCHICAL MULTILINGUAL MODEL WITH DATA SAMPLING

The neural network architecture presented in this paper is hierarchical and modeled after the topology proposed in [11]. It combines the multilingual training strategy from [30] with the stacked DNN structure from [31]. The hierarchical DNN based model with ML representations is illustrated in Figure 1. The two DNNs in this stacked architecture have a similar structure with 5 layers comprising of 1024 sigmoid units each, except for the bottleneck layer, which has 80 sigmoid units, and a final soft-max layer.

As illustrated in Fig. 1, the input layer to the first DNN are 40-dimensional log-mel filter bank features spliced together with a context +/-5 frames. The second DNN uses the 80-dimensional bottleneck features extracted from the first DNN. The context is expanded to include 10 frames on each side and then subsampled at a five-frame interval to produce a 400-dimension input vector for the second DNN. Both DNNs, use independent softmax output layers corresponding to each of the 10 training languages used: Assamese, Bengali, Pashto, Turkish, Tagalog, Vietnamese, Haitian Creole, Lao, Tamil and Zulu. These languages cover the languages used in the Base and OP1 evaluation periods of the Babel program [9]. We used the development data from the Assamese language as the held-out set to determine the stopping criterion for training this multilingual network. While it is possible to have a fully connected final layer across the targets of all languages, we choose this representation to allow for faster training of the network resulting from fewer parameters in the last layer. All hidden layers are shared across all languages, allowing the network to learn a truly multilingual representation. The output targets for each of the languages are the context dependent states derived from unilingually trained, speaker adapted decision trees. However, given these languages were processed during different phases of the program, we simply reused the states that were generated then, with alignments generated from either GMMs or DNNs.

Our multilingual representations on the target language are derived from the bottleneck layer of the second DNN shown in Fig. 1, by passing the target language through the multilingual network. In this paper, we focus on ML representations obtained with no fine-tuning on the target language, i.e., the multilingual network does not see any of the target language data.

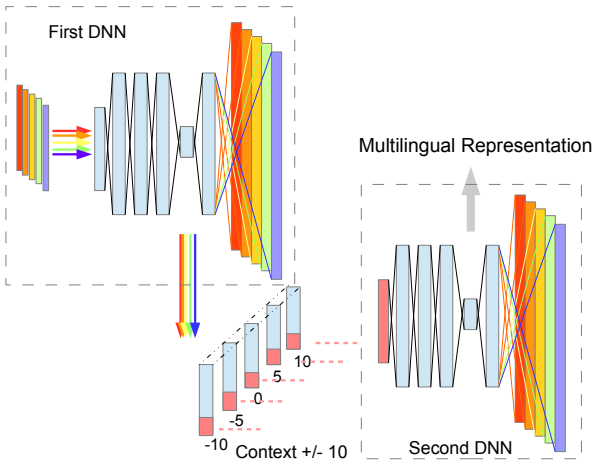

Fig. 1. *IBM Hierarchical Multilingual DNN.*

2.1. Data Sampling

In order to speed up training of ML representations, we propose a data sampling strategy that allows the network to see only a fraction of the data in each epoch. With several training epochs, the network eventually sees all of the training data.

The training data from each language is organized into 30 sets, with each set comprising of several mini batches. During each training epoch, for each language, a fraction, r of the training data is used. It is possible for some sets to be used more often than others during the training process. The model is trained with 15 epochs of Stochastic Gradient Descent(SGD) on a single NVIDIA K40x GPU.

Table 1 lists the converged cross-entropy objective function values on the held-out set, and the corresponding times to train the networks for different sampling ratios r. We explore different sampling ratios for training both DNNs in the hierarchical architecture. When a sampling ratio of $1/6$ is used for both DNNs (IBM1) i.e., both networks train on only one-sixth of the overall training data during each epoch, the fastest training time of four days is observed. A slightly better convergence is obtained when the second network in the stacked architecture is allowed to train on half the training data per epoch (IBM2). However, the training time increases proportionately by 50% relative. Using the entire training set per epoch for training the second net, does decrease the objective slightly further, but at the expense of a much higher training period (10 days). If the entire training data (IBM) is used at every epoch to train both nets, the training time increases drastically to three weeks. However, the objective function value achieved with all the data being used for both nets is only slightly better (1.12) compared to using half the training data for the second DNN and $1/6^{th}$ for the first net(1.17). Our experiments also showed that a single net is only able to bring the objective down to 1.57 when trained with $1/6th$ of the data per epoch. Even when using the entire training data per epoch, only a very small improvement in objective, to 1.47 can be achieved, while the training time increases drastically by a factor of three.

As a sanity check, we compared the objective values presented in Table 1 on the held-out set against the converged objective from a unilingual DNN model (1.23) trained with the FLP data from the

language of the held-out set, i.e. Assamese. The hierarchical multilingual net converges to a better objective (1.12) than the baseline.

DNN.1	DNN.2	XENT	Training time (Days)
$r = 1/6$(IBM1)	$r = 1/6$	1.21	4
$r = 1/6$(IBM2)	$r = 1/2$	1.17	6
$r = 1/6$(IBM3)	$r = 1$	1.15	10
$r = 1$(IBM)	$r = 1$	1.12	21

Table 1. *ML training with various sampling ratios.*

3. IARPA BABEL OP2 SURPRISE LANGUAGE EVALUATION

The work reported in this paper is focused on the IARPA Babel OP2 surprise language (Swahili) evaluation. There are three evaluation scenarios based on the amount of transcribed training data, namely, Very Limited Language Pack (VLLP), Active Language Pack (ALP) and Full Language Pack (FLP). In the VLLP case, the training data comprises of only 3 hours of transcribed audio. Particpants have only two weeks to train ASR and KWS systems. In the ALP case, one hour of transcribed audio is provided initially. However, participants are allowed to automatically select two additional hours of audio using the one-hour set as a seed, for which manual transcripts will be provided [32–37]. Both VLLP and ALP allow the use of 40 hours of un-transcribed audio. VLLP and ALP conditions serve as two different methods to arrive at the best performing ASR and KWS systems when very little transcribed data ia availble. The algorithms developed for these two conditions are contrasted with the FLP scenario, wherein, 40 hours of transcribed audio is available. The evaluation period runs for a week, with 70 hours of audio to be searched in all three scenarios. The keywords are the same across all conditions. Textual data to derive lexicons and language models are derived from webcrawls [38] and common across all participants. For the evaluation itself, ML representations and use of webcrawls are allowed for VLLP and ALP conditions only. However, for this study, in order to obtain a better understanding of ML representations, we compare their value across all three conditions.

The VLLP training data contains 3K sentences (28K words) with a vocabulary size of 5K. In contrast, the FLP training data contains 50K sentences (353K) words with a vocabulary size of 24K. The *development* data comprises of 15 hours of audio (11K sentences) and 4k query terms [39] and is the same across all three conditions. An internal *tuning* set of 3 hours of audio (3.5K sentences) was used to tune the hyperparameters of ASR and KWS systems.

The ALP evaluation condition is the selection of two hours worth of segments from the untranscribed pool for transcribing, and then building models given the initial one-hour plus the additional two hours of transcribed audio. The untranscribed pool is initially segmented at silence regions, followed by an entropy-based selection of segments. The entropy for each segment is computed using a grapheme probability density function computed over the consensus network. Segments are selected in a round-robin fashion by speaker; the segment having the highest grapheme-based entropy for a given speaker is chosen as we rotate through speakers.

The analyses of ASR and KWS systems reported throughout this paper are on the *tuning* set. The final tuned systems are then evaluated on the development set (Section 6).

3.1. IBM ASR System

The baseline speaker-independent (SI) acoustic model used in IBM's ASR system is described below. The input features are 13-dimension PLP features with speaker-based mean and variance normalization. A context of 9 frames is spliced together and projected to a 40-dimensional feature space using linear discrimant analysis(LDA), and the class-conditional distributions are further diagonalized using a global, semi-tied covariance(STC) transform.

In the SI ML pipeline, the above PLP+LDA+STC features are fused with ML features, transformed by LDA and STC, and then used as input for a two-fold DNN pipeline. In each fold, a new alignment is generated with the current model and a new decision tree is built on top the alignment. The DNN training procedure comprises of: (1) discriminative layer-wise pretraining [40], (2) training with cross-entropy criterion and (3) training with the state-level minimum Bayes risk(sMBR) criterion [19, 20]. The DNN comprises of 3 hidden layers of 1024 ReLU units, followed by one 1024-unit sigmoid layer and a 128-unit linear layer. In the SA ML systems, ML features spanning a context of 9 frames are spliced together and projected down to a 40 dimension feature space with LDA+STC, followed by a constrained MLLR transform [41]. All DNN models used in this paper are hybrid models [20]. The IBM Attila speech recognition toolkit [42] is used for training the models presented in this paper.

The baseline language models (LM) are Kneser-Ney (KN)-smoothed bigram models. For FLP condition, the vocabulary size was $24K$ and for the VLLP/ALP conditions, the vocabulary size was $5K$.

3.2. On-the-fly Lattice KWS

We use an on-the-fly version of lattice based keyword search to generate our KWS results. The queries are read in and processed to create query Finite State Transducers (FSTs). In-vocabulary (IV) queries are represented at both the token (word or syllable) level and the grapheme level, and query expansion is applied to the grapheme FSTs using a confusability model. OOV queries are only represented at the grapheme level, and have the same degree of query expansion as the IV queries. Next, as ASR is performed for each segment, a lattice is generated in memory, converted to a weighted FST index, the queries are searched for in the index via composition, and any hits are recorded. When ASR is finished, the results are written to disk in the form of postings lists for the token and grapheme searches. Finally, the postings lists are merged in a cascaded fashion: if any token results for a query are present, they are used; otherwise, the grapheme results are used. It is important to note that the output of this on-the-fly KWS is identical to what would be produced by a standard FST based KWS [43, 44] system that writes out lattices, compiles them into indexes, and then runs search. The primary advantage of the on-the-fly approach is that we avoid the need to write out the lattices, which can be extremely large.

We used cleaned webcrawls to augment the ASR dictionary, vocabulary and LM for KWS. After addition of web crawls the vocabulary size of the language models increases to roughly $350K$ for all three conditions. This reduced the OOV rate of KWS queries by nearly 76% relative on the VLLP and the ALP data set and 64% on the FLP set. The KWS results presented in this paper are based on word lattices and query expansion of $1000-nbest$ applied for phonetic search. The performance of the KWS system is measure using the Maximum Term Weighted Value (MTWV) metric described in [45].

4. MULTILINGUAL REPRESENTATIONS FOR SWAHILI

This section describes multilingual representations for Swahili ASR and KWS. This section details the experiments with ML representations from CUED and RWTH. The DNN used to derive the CUED ML features is similar to [11]. The input features are 24-dimensional log Mel magnitude spectrum filter banks, pitch, probability of voicing, and their derivatives. The RWTH ML features are described in [14] and include long-span features. Both RWTH and CUED multilingual networks are trained on 11 languages.

4.1. ML Representations in Low Resource Conditions

4.1.1. Data Sampling

Based on the results in Table 1, and the evaluation time constraints, we selected the ML representations, IBM1 and IBM2 for further exploration in the VLLP condition, and IBM2 and IBM for the ALP condition. The evaluation for the three conditions was staged with ALP following VLLP. Ideally, we would have liked to use the IBM ML representation from the last row of Table 1 for the VLLP condition but could not do so due to the time constraints of the evaluation.

To illustrate the impact of ML representations derived from different sampling ratios, we select the following configurations. For the VLLP condition, the ML features are fused with SI, PLP+LDA+STC features and used as input features to train a 4-layer DNN. The targets for this DNN are 1000 context-dependent states from a DNN-alignment based decision tree. For the ALP condition, the ML features were speaker-adapted and fused with a second set of ML representations obtained from RWTH. Table 2 presents the ASR systems' performance on the surprise language, Swahili, when using these ML representations. It can be seen from the table that in both conditions, ML representations are able to derive better hidden language representations, if the multilingual nets see more data in each training epoch.

	IBM1	**IBM2**	**IBM**
VLLP	65.1	64.3	—
ALP	—	60.3	59.8

Table 2. *ASR performance with ML features generated from different sampling ratios.*

4.1.2. VLLP

In this section, we compare the ASR performance of four different configurations that use SI and SA, ML representations. Table 3 lists WERs on the *tuning* set with different ML features at intermediate training steps of the recipe presented in Section 3.1. In Table 3, 'XE' refers to the training step with cross-entropy as the objective function and 'sMBR' refers to the sequence training step. The two folds of DNN training referred to in Section 3.1 are denoted by suffix '1' and '2' respectively. First, we observe as the models are refined with realignments during the training steps, the ASR performance improves (illustrated in Rows 1 through Row 4), regardless of the type of ML feature used. The gain in performance for each of the ML features, ranges from 2.5% to 4.0% absolute (Columns 1 thru 3). The second observation is with regards to the complimentarity of different ML representations. The last column in Table 3, is a configuration that fuses two different ML features, the speaker-adapted IBM2 features from IBM with the SI ML features from RWTH. A reduction in WER of 2.0% absolute is seen with these combined features as well, as the models are refined during various stages of training, illustrating the complimentarity of these ML representations. It can also be seen that this type of a gain holds through the various intermediate training stages. Third, we observe that re-alignments with the first set of models helps in decreasing the WER further, yielding gains in the range 0.4% to 1.9% (Compare rows sMBR.1 and sMBR.2) absolute across the different configurations. Last, we observe that the ML representations from the various sites are comparable and converge to more or less the same WER (Row 4), with the ML features from RWTH outperforming the other two ML features.

Stage	**RWTH-SI**	**CUED-SI**	**IBM2-SI**	**IBM2-SA +RWTH**
XE.1	65.7	66.9	68.3	63.3
sMBR.1	63.3	64.8	66.2	62.1
XE.2	63.5	65.3	66.1	62.2
sMBR.2	62.9	64.4	64.3	61.3
sMBR.2 MTWV	0.4102	0.4197	—	0.4783

Table 3. *Performance of ML features on Swahili VLLP.*

4.1.3. Semi-supervised Learning

Semi-supervsed learning (SSL) has shown to be beneficial to ASR and keyword search in the Base and OP1 evaluation periods [46]. Motivated by these previous results, the untranscribed data is first decoded with an initial VLLP model trained on just the transcribed data, and subsequently merged to form a unified, larger training data set. The ML representations are derived on this larger data set and used to train the final DNN on the target language. With the addition of training data, the number of output targets is increased from 1000 to 3000.

Table 4 illustrates the ASR performance using SI ML features from RWTH and CUED, with and without SSL training under the VLLP condition. SSL yields 2.6% to 3.4% absolute reduction in WER. This also provides a 6.3% to 8.7% relative increase in MTWV.

	w/o SSL	**w/ SSL**
	WER/MTWV	**WER/MTWV**
RWTH-SI	62.9/0.4102	60.3/0.4458
CUED-SI	64.4/0.4197	61.0/0.4461

Table 4. *Comparison of Swahili VLLP performance with and without semi-supervised learning.*

Table 5 demonstrates the performance of SA ML representations with SSL. The PLP-SA row refers to SSL applied to a DNN trained on speaker-adapted PLP features and results in a WER of 62.4% on the *tuning* set. RWTH-SA refers to a DNN (described in Section (4.1) trained on the speaker-adapted ML features from RWTH. This model results in a WER of 60.0%. Fusion of these ML features with the IBM IBM2 ML representations, provides a further reduction in WER of 1.3% absolute. Addition of a third speaker-adapted ML feature, from CUED (last row in the table) reduces the WER further by another 1.0% absolute. The three different ML representations are clearly complimentary resulting in a 2.3% absolute reduction in WER over using the best single ML representation. When compared to a DNN trained with SI, ML features from RWTH (See Table 3), we observe that multiple ML features in conjunction with

SSL reduced WER by 5.2% absolute, from 62.9% to 57.7%. It is interesting to note that an SI, ML representation based DNN with no SSL(62.9% from Table 4), matches the performance of a DNN trained with simple, speaker-adapted PLP features in Table 5. This implies that ML representations do capture acoustic representations well, i.e., ML features from 3 hours of transcribed data on the target language can achieve the same level of ASR performance as PLP features from 3 hours of transcribed data and approximately 40 hours of untranscibed data. The last row in this table refers to a data augmentation technique originally presented in [47]. Here, the speaker-adapted ML features from RWTH and the training data is increased 8-fold. Interestingly, this model yields the same ASR performance as the fused ML representations in Row 4, suggesting that ML representations do capture the acoustics of the target language very well.

Features	WER	MTWV
PLP-SA w/o ML	62.4	0.4715
RWTH-SA	60	0.4684
RWTH-SA + IBM2-SA	58.7	0.4703
RWTH-SA + IBM2-SA + CUED-SA	57.7	0.4809
RWTH-SA + 8xdata augmentation	58.7	—

Table 5. *Impact of several ML features on Swahili VLLP with semi-supervised learning.*

4.1.4. ALP

Table 6 shows the comparison of various ML features used for the ALP evaluation scenario. As mentioned in Section 2.1, the IBM ML representation (IBM) is used for training a DNN on the target language.The first three rows of the table present ASR performance when three different SI ML features are used. Similar to the VLLP evaluation condition, the different ML representations are very similar in performance. Speaker-adaptive transformation applied to the IBM ML features yields a reduction of 0.9% WER absolute. This finding is consistent with the VLLP condition, where similar gains were observed (See Table 3). Feature combination with RWTH ML features gives an additional 1.1% reduction in WER; and SSL an additional 2.4% reduction in WER, This is consistent with our previous observation for the VLLP condition. The use of multiple ML representations and SSL (last row) accounts for a 4% absolute reduction in WER over a single ML feature (Row 4). RWTH-SI features.

Models	WER	MTWV
RWTH-SI	61.5	—
CUED-SI	63.4	—
IBM-SI	61.8	0.4454
IBM-SA	60.9	0.4669
IBM-SA + RWTH	59.8	0.4823
IBM-SA + SSL	58.4	-
IBM-SA + RWTH + SSL	57.4	0.4714

Table 6. *Performance of ML features on Swahili ALP.*

4.2. ML Features on Swahili FLP

In the earlier sections, we demonstrated the significant impact of ML representations for the low-resource conditions. In this section, we explore its use for the FLP scenario with 40 hours of transcribed data. The baseline DNN is trained on speaker-independent PLP features using the recipe outlined in Section 4.1 and yields a WER of 50.9% on the development data set (See Table 7). The use of speaker-adapted PLP features decreases the WER further to 49.0%. The addition of IBM ML features (IBM) to the SA-PLP features results in a significant reduction of WER by 4.2% absolute. This strong result highlights the value of ML representations even when 40 hours of transcribed data is available in the target language.

Stages	Baseline
NoML-SI	50.9
NoML-SA	49.0
NoML-SA + IBM	44.8

Table 7. *Performance of ML features on Swahili FLP.*

5. FINE TUNING OF ML FEATURES ON THE TARGET LANGUAGE

In this section, we investigate the value of refining the ML representations with an additional training pass using the available data from the target language. We use the ALP evaluation scenario for this study. In the configuration presented here, the parameters of the second DNN in the stacked architecture were adjusted on the target language. The last layer of the second DNN is randomly initialized with the output targets set to the context-dependent states of the target language. The remaining layers are initialized with the same weights obtained from the multilingual training.

Table 8 presents the cross-entropy values for different training configurations that correspond to a different set of layers of the DNN being updated. The WERs presented in this table are a result of hybrid decoding using this refined DNN directly. The first three rows correspond to refining the weights of the last layer, weights starting from the second hidden layer onwards and all layers of the DNN respectively. A significant reduction in the objective and WER is obtained when more layers of the network are tuned to the target language.

Layers	XENT	WER
5	3.63	69.2
3+	2.52	—
2+	2.54	62.5
all	2.56	—

Table 8. *Cross-entropy and WER on Swahili ALP after fine-tuning.*

The network from which ML representations are derived is fine tuned from the second hidden layer onwards. A DNN on the three hours of ALP transcribed data was trained using the recipe in Section 4.1. Table 9 captures the WER on the *tuning* set at intermediate training steps. Even though the fine tuned features outperform the vanilla ML features at the early stages, the gains gradually disappear in the subsequent training steps.

6. KEYWORD SEARCH ANALYSIS

In this section we analyze the impact of ML features on our KWS results obtained using the on-the-fly KWS described in 3.2. We compare the MTWV achieved by systems trained with multilingual features and systems trained without multilingual features on all three

Stages	No finetune	Finetune
XE.1	65.5	64.4
sMBR.1	63.3	62.9
XE.2	63.0	62.8
sMBR.2	61.8	61.8

Table 9. *Comparison of ML features with and without fine tuning on Swahili ALP.*

BABEL conditions: ALP, VLLP and FLP. The comparisons were done on the tuning set for ALP and VLLP conditions, and the development set for the FLP condition. Table 10 provides the tuning set results for ALP and VLLP. We observe that ML features give consistent gains on both the IV and OOV terms. We obtain a relative MTWV improvement of 5.8% for ALP and 2% for VLLP. The development set MTWV results for all three conditions are reported in Table 11. We obtain a relative MTWV improvement of 4.2% for ALP, 7.6% for VLLP, and 6.7% for FLP. Although the results presented in Table 10 and Table 11 are obtained with word lattices, similar trends hold for our morph and syllable-based KWS systems.

System	ML feats	MTWV		
		IV	OOV	Total
ALP	Yes	0.4337	0.5747	0.4823
ALP	No	0.4078	0.5470	0.4559
VLLP	Yes	0.4335	0.5638	0.4809
VLLP	No	0.4105	0.5787	0.4715

Table 10. *Comparison of MTWV between a multilingual system and a unilingual system trained only on Swahili data for ALP and VLLP on the tuning set. The table also includes the MTWV breakdown for IV and OOV queries defined according to the original non-web vocabularies.*

System	ML feats	MTWV		
		IV	OOV	Total
ALP	Yes	0.4708	0.5283	0.4946
ALP	No	0.4490	0.5104	0.4745
VLLP	Yes	0.4870	0.5071	0.4957
VLLP	No	0.4430	0.4870	0.4605
FLP	Yes	0.5780	0.5100	0.5736
FLP	No	0.5413	0.4780	0.5374

Table 11. *Comparison of development set MTWV between a multilingual system and a system trained only on Swahili data for ALP,VLLP and FLP conditions. The table also includes the MTWV breakdown for IV and OOV queries.*

Figure 2 shows the variation of MTWV with query length measured by number of graphemes using systems with and without multilingual features for the three conditions - FLP, VLLP and ALP. We observe that the use of multilingual features helps bridge the gap between the performance for the data-rich FLP and the data-sparse VLLP/ALP conditions. Multilingual features give consistent KWS performance gains for all three conditions. We also note that KWS performance increases with query length. This is because short queries are usually more acoustically confusable than longer queries.

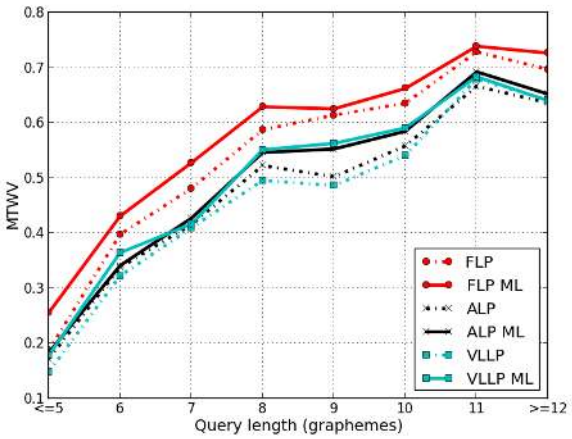

Fig. 2. *This figure shows the variation of MTWV with query length (number of graphemes) for the three conditions (FLP, VLLP and ALP) using systems with and without multilingual features.*

7. CONCLUSIONS

Multilingual acoustic representations proved to be crucial for building systems under the strict resource and time constraints of the OpenKWS15 Evaluation. Using multilingual representations significantly improved our ASR and KWS performance (relative 9% for WER and 5% for MTWV) This paper presented our findings in the process of building these systems which can be summarized as follows

- The data sampling strategy presented in the paper can speed up the training of multilingual representations without much loss in performance.
- Fusion of diverse multilingual representations yields substantial ASR and KWS gains.
- Fine-tuning the multilingual representations on the target language (Swahili) did not improve performance.
- Speaker adaptation and data augmentation of these representations improved word-error rate (WER) and KWS performance
- Incorporating un-transcribed data through semi-supervised learning, improves WER and KWS performance.
- Multilingual features were helpful even when forty hours of transcribed audio in the target language is available.

The final KWS submission to the OpenKWS15 evaluation was a combination of multiple systems using multilingual representations developed at IBM, RWTH, and CUED. It resulted in an MTWV of 0.5888 for VLLP and 0.6020 for ALP on the tuning set. These KWS systems yielded the best ATWVs[1] on the evaluation data across all three conditions - ALP: 0.5952, VLLP: 0.5797 and FLP: 0.6548. It is important to note that an ATWV of 0.3 is considered acceptable and is the program goal for Babel OP2.

[1] ATWV is the TWV at the submitted operating point.

8. REFERENCES

[1] Alex Waibel, Hagen Soltau, Tanja Schultz, Thomas Schaaf, and Florian Metze, "Multilingual speech recognition," in *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 33–45. Springer, 2000.

[2] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 7319–7323.

[3] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*. IEEE, 2013, pp. 7304–7308.

[4] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 6704–6708.

[5] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. SLT*, 2012, pp. 246–251.

[6] D. Imseng, H. Bourlard, and P. N. Garner, "Using KL-divergence and multilingual information to improve ASR for under-resourced languages," in *Proc. ICASSP*. IEEE, 2012, pp. 4869–4872.

[7] L. Burget, P. Schwarz, M. Agarwal, Pinar Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrow, R. C. Rose, and S. Thomas, "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *Proc. ICASSP*. IEEE, 2010, pp. 4334–4337.

[8] František Grézl and Martin Karafiát, "Combination of multilingual and semi-supervised training for under-resourced languages," in *Proc. Interspeech*, 2014.

[9] Mary P. Harper, "http://www.iarpa.gov/index.php/research-programs/babel," .

[10] Z. Tuske, R. Schluter, and H. Ney, "Multilingual hierarchical MRASTA features for ASR," in *Interspeech*, 2013, pp. 2222–2226.

[11] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014, pp. 7904–7908.

[12] K. Knill, M.J.F. Gales, S. Rath, P. Woodland, C. Zhang, and S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *ASRU*, 2013.

[13] Florian Metze, Xavier Anguera, Etienne Barnard, Marelie Davel, and Guillaume Gravier, "The spoken web search task at MediaEval 2012," in *Proc. ICASSP*. IEEE, 2013, pp. 8121–8125.

[14] Christian Plahl, Ralf Schlüter, and Hermann Ney, "Hierarchical bottle neck features for lvcsr," in *Interspeech*, 2010.

[15] Fabio Valente and Hynek Hermansky, "Hierarchical and parallel processing of modulation spectrum for asr applications," in *ICASSP*, 2008.

[16] Mark J. F. Gales, Kate M. Knill, Anton Ragni, and Shakti P. Rath, "Speech recognition and keyword spotting for low resource languages: Babel project research at CUED," in *Spoken Language Technologies for Under-Resource Languages*, 2014.

[17] Haipeng Wang, Anton Ragni, Mark JF Gales, Kate M Knill, Philip C Woodland, and Chao Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Proc. Interspeech*, 2015.

[18] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 8619–8623.

[19] J. Martens, "Deep learning via hessian-free optimization," in *ICML*, 2010.

[20] Brian Kingsbury, Tara N Sainath, and Hagen Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Interspeech*, 2012.

[21] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al., "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[22] Oriol Vinyals and Daniel Povey, "Krylov subspace descent for deep learning," *arXiv preprint arXiv:1111.4259*, 2011.

[23] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu, "1-Bit Stochastic Gradient Descent and its Application to Data-Parallel Distributed Training of Speech DNNs," in *Proc. Interspeech*, 2014.

[24] Tara N Sainath, I-hsin Chung, Bhuvana Ramabhadran, Michael Picheny, John Gunnels, Brian Kingsbury, George Saon, Vernon Austel, and Upendra Chaudhari, "Parallel Deep Neural Network Training for LVCSR Tasks using Blue Gene/Q," in *Proc. Interspeech*, 2014.

[25] Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal, "On the use of stochastic hessian information in optimization methods for machine learning," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 977–995, 2011.

[26] Tanja Schultz and Katrin Kirchhoff, *Multilingual speech processing*, Academic Press, 2006.

[27] A Ragni, MJF Gales, and KM Knill, "A language space representation for speech recognition," in *Proc. ICASSP*, 2015.

[28] Hynek Hermansky and Nelson Morgan, "Rasta processing of speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 578–589, 1994.

[29] Hynek Hermansky and Petr Fousek, "Multi-resolution rasta filtering for tandem-based asr," Tech. Rep., IDIAP, 2005.

[30] Stefano Scanzio, Pietro Laface, Luciano Fissore, Roberto Gemello, and Franco Mana, "On the use of a multilingual neural network front-end," in *Interspeech*, 2008.

[31] Frantisek Grezl, Martin Karafiat, and Lukas Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. Interspeech*, 2009, pp. 2947–2950.

[32] Nancy F. Chen et al, "Low-resource keyword search strategies for tamil," in *Proc. ICASSP*, 2015.

[33] Dilek Hakkani-Tur, Giuseppe Riccardi, and Allen Gorin, "Active learning for automatic speech recognition," in *Proc. ICASSP*, 2002.

[34] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[35] Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion," *Computer Speech & Language*, vol. 24, no. 3, pp. 433–444, 2010.

[36] Nobuyasu Itoh, Tara N Sainath, Dan Ning Jiang, Jie Zhou, and Bhuvana Ramabhadran, "N-best entropy based data selection for acoustic modeling," in *Proc. ICASSP*, 2012, pp. 4133–4136.

[37] Yi Wu, Rong Zhang, and Alexander Rudnicky, "Data selection for speech recognition," in *Proc. ASRU*, 2007, pp. 562–565.

[38] Gideon Mendels, Erica Cooper, Victor Soto, Julia Hirschberg, Mark Gales, Kate Knill, Anton Ragni, and Haipeng Wang, "Improving speech recognition and keyword search for low resource languages using web data," in *Proc. Interspeech*, 2015.

[39] Jia Cui, Jonathan Mamou, Brian Kingsbury, and Bhuvana Ramabhadran, "Automatic keyword selection for keyword search development and tuning," in *Proc ICASSP*. IEEE, 2014, pp. 7839–7843.

[40] Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novk, and Abdel rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011, pp. 30–35.

[41] M. J. K. Gales, "Maximum-likelihood linear transforms for hmm-based speech recognition," 1998, pp. 75–98.

[42] Hagen Soltau, George Saon, and Brian Kingsbury, "The IBM Attila speech recognition toolkit," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 97–102.

[43] Cyril Allauzen, Mehryar Mohri, and Murat Saraclar, "General indexation of weighted automata - application to spoken utterance retrieval," in *Proceedings of HLT*, 2004.

[44] Brian Kingsbury, Jia Cui, Xiaodong Cui, Mark JF Gales, Kate Knill, Jonathan Mamou, Lidia Mangu, David Nolden, Michael Picheny, Bhuvana Ramabhadran, et al., "A high-performance Cantonese keyword search system," in *Proc. ICASSP*. IEEE, 2013, pp. 8277–8281.

[45] J Fiscus, J Ajot, and G Doddington, "The spoken term detection (std) 2006 evaluation plan," *NIST USA, Sep*, 2006.

[46] Jia Cui, Bhuvana Ramabhadran, Xiaodong Cui, Andrew Rosenberg, Brian Kingsbury, and Abhinav Sethy, "Recent improvements in neural network acoustic modeling for LVCSR in low resource languages," in *Interspeech*, 2014, pp. 840–844.

[47] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *Proc. ICASSP*. IEEE, 2014, pp. 5582–5586.