MULTILINGUAL DATA SELECTION FOR TRAINING STACKED BOTTLENECK FEATURES

Ekapol Chuangsuwanich, Yu Zhang, James Glass

MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA

{ekapolc, yzhang87, glass}@mit.edu

ABSTRACT

Deep Neural Networks (DNNs) trained on multilingual data have proven useful for improving speech recognition in languages with limited resources. In this framework, data from rich resource languages are pooled together to train a single system and then adapted to a new language. However, data from a rich language that are similar to the target language are generally more helpful. We explore methods of training bottleneck features by using data that are more similar to the target language. Our experiments on speech recognition and keyword spotting tasks with IARPA-Babel languages show that our proposed methods outperform typical multilingual DNNs.

***Index Terms*—** Multilingual, Bottleneck features, DNN, Data selection

1. INTRODUCTION

Automatic Speech Recognition (ASR) has been receiving considerable exposure recently. However, ASR capabilities are available for less than 2% of the languages spoken around the world. This is because traditional ASR development requires significant linguistic resources in the form of annotated data for acoustic and language modeling, and pronunciation dictionaries that are expensive and time-consuming to produce. Given that ASRs perform best with hundreds or thousands of hours of speech data, it is challenging to obtain good performance with limited resources.

To address the resource limitation issues, many researchers are exploring the use of out-of-domain acoustic resources, such as multilingual corpora. Approaches such as in [1, 2] try to learn a common lower-dimensional subspace across languages in order to reduce the amount of parameters that need to be learned for a new language. Another popular approach is multi-task training using DNNs. In a multi-task setup, a single DNN is trained to generate outputs for multiple languages with some tied parameters. This approach has been used as a robust feature extraction via bottleneck (BN) features [3, 4, 5, 6] or as classifiers in hybrid DNN-HMM approaches [7, 8].

When multilingual resources are rich and diverse, such as in the IARPA-Babel program, one question that arises is how to best take advantage of the resources. Although it can be beneficial to use more of the available languages [9], there is also evidence that a source language that is close to the target language is more beneficial than a random one [10]. In our previous work [4], we proposed a method for adapting a multilingual DNN that can exploit such information. We automatically identified the closest source language by the use of Language Identification (LID), and then use it to re-train part of a BN network before adapting to the target language.

The work presented here is an extension of our previous work in [4]. We propose a method to identify a portion of the multilingual data that is closer to the target and, thus, more beneficial for multilingual adaptation. Experiments on the IARPA-Babel corpora show that BN systems trained using the closest frames provide gains in both ASR and keyword spotting. We also provide further analysis on the usefulness of the DNN used for LID.

2. STACKED BOTTLENECK ARCHITECTURE

The BN features used in this work follow our previous work in [11]. A SBN is a hierarchical architecture realized as a concatenation of two DNNs each with its own bottleneck layer. The outputs from the BN layer in the first DNN are used as the input features for the second DNN, whose outputs at the BN layer are then used as the final features for standard GMM-HMM training.

2.1. Multilingual training of SBN features

There are several methods for training multilingual DNNs. In [12, 13], a multilingual phoneset is created, and all the phonemes from the source languages are mapped to the set. The work in [13, 14] shows that a simpler scheme of concatenating each language outputs in the softmax layer can perform just as well. Furthermore, when concatenating language outputs, normalizing the softmax layer individually within each language during training will yield slightly better results [5, 9]. This training technique has also been applied successfully to multilingual hybrid DNNs in [7, 8], and as a feature extractor for hybrid DNNs [15]. In this work, we use this method for training the multilingual DNN since it does not require mapping of the phonesets and still provides state-of-the-art results.

When adapting the multilingual DNN to a new language, i.e., the target language, there often exists a limited amount of training data in that language. Adapting the multilingual DNN using data from the target language gives additional gain over the purely multilingual DNN. For hierarchical architectures, such as SBN, our previous work in [4] and an independent investigation in [10] seem to suggest that the two DNNs in the SBN architecture behave differently in terms of adaptation. The first DNN extracts more language independent cues from the acoustics, while the second DNN is more language dependent and is more phonetically oriented. Our previous work [4] shows that using just the language closest to the target language to train the second DNN can outperform the multilingual second DNN. Thus, in this paper we will focus mainly on the training and adaptation of the second DNN.

Supported in part by the Intelligence Advanced Research Projects Activity(IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. 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

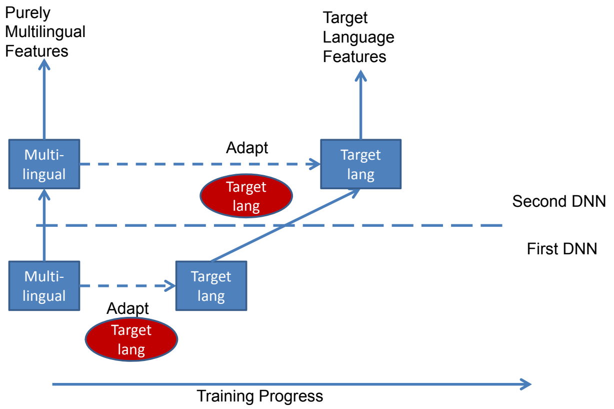

Fig. 1. Steps to adapt a multilingual SBN to a new language.

A flowchart of how a multilingual SBN is adapted to the target language is shown in Fig. 1. The first DNN is adapted to the target language by applying additional fine-tuning using the data from the target language. The softmax is replaced by the target language state labels with random initialization, while the hidden layers are initialized from the multilingual DNN. After the first DNN converges, the same procedure is applied to the second DNN.

3. LID FOR MULTILINGUAL TRAINING

3.1. Training from the closest source language

In [4] we devised a scheme for selecting better source languages to train the SBN by relying purely on the acoustics. This can be done by training a LID system from the pool of source languages using DNNs. The output labels of the DNN are the language tags, while the input are the stacked frames used as the input to the SBN. Unlike typical DNN-based LID work, such as in [16, 17], we chose to use the same input features as used in the SBN because we want to ensure that the LID DNN decides which languages are similar based on what the SBN would observe.

To identify the closest source language, we first train a LID DNN to classify source languages. Given N source languages, we train the DNN with $N + 1$ output labels: One label for each language, and an additional label for silence (SIL). SIL includes actual silence, noise, cough, and laughter from every source language. The SIL label is included to exclude non-language specific sounds from the scoring. The language that is closest to the target language can be identified by computing the average of posterior scores over all frames from the target language. Note that sometimes the language closest to the source language in the LID sense might not align with linguistic knowledge. This can be due to channel and other non-speech effects.

To train the LID-based system, We start in the same manner as the multilingual method by adapting the first DNN with the target language. However, instead of using the second DNN of the multilingual to initialize, we train the second DNN from random initialization using the closest language's data and output targets. While it is usually the case that the target language does not have enough data by itself to train a DNN from scratch, this should not be the case for the source language[1]. Training from scratch is usually preferable because the multilingual DNN is trained on data that could contain irrelevant information with respect to the target language. A final adaptation step is then done using the target language.

3.2. Frame selection

Data selection has been explored in many contexts, such as for semi-supervised training where a portion of untranscribed data is selected to re-train a speech recognizer [18, 19], or for active learning where the goal is to select the smallest subset of data to transcribe that maximizes performance [20, 21, 22]. Our situation differs from these prior approaches in that we want to select subsets of (transcribed) data that are close to the target language. Thus, we do not select for maximal variability. Also, since we have limited amounts of target data, we prefer not to use a speech recognizer in the selection process. Finally, we select at the frame level, rather than the utterance level to maximize closeness at the phonemic level only.

In [4] we observed that selecting the closest subset from a particular source language can sometimes be more beneficial than using all of it. This offers some explanation to why the multilingual SBN can perform worse than the closest language setup. However, it has always been observed that in a multilingual setting, having more source languages usually helps due to better coverage of phonemes and acoustical phenomena, as well as the simple fact that there is more data [10, 12]. From these observations we propose an improvement over the closest language training scheme by selecting frames from all source languages that are closest to the target language to train the second DNN.

To select the closest frames from the multilingual pool, we need a way to score and rank all the frames. We can do so by training N frame selection DNNs, one for each source-target pair. Each frame selection DNN is a two-class DNN where the training data are the frames from the source and target languages with their corresponding language labels. The score of a frame from any source language is then the posterior probability of that frame coming from the target language computed by using the corresponding DNN. Although each frame selection DNN is trained independently from the rest of the source languages, we observe that the distribution in the rankings of all source language frames correlates well with the scores given by the LID DNN; the languages with higher LID scores have more highly ranked frames.

We train N frame selection DNNs for the ranking instead of one single DNN with $N + 2$ output labels (the sources, SIL, and the target) because the existence of a close language pair in the source pool can skew the ranking of the frames. For example, consider the case when the source languages are Assamese, Bengali, and Zulu, and the target language is Telugu. Assamese, Bengali, and Telugu are all Indian Languages, so we expect the frames from the Assamese and Bengali to have higher probabilities of being Telugu than frames from Zulu. However, since Assamese and Bengali are very similar languages (more similar together than to Telugu), the posterior probability for an Assamese frame will mostly be biased towards Bengali. On the other hand, a Zulu frame would have no such effect and may have a higher posterior for Telugu.

After selecting the closest frames, the training procedure follows the closest language method discussed in Section 3.1.

4. EXPERIMENTS

4.1. IARPA-Babel corpus

The IARPA-Babel program focuses on ASR and spoken term detection on low-resource languages [23]. The goal of the program is to reduce the amount of time needed to develop ASR and spoken term detection capabilities in a new language. The Babel corpus consists of collections of speech from a growing list of languages. For this work we will consider the Full pack (FLP) of 11 languages released

[1] We can always initialize from the multilingual DNN and adapt to the closest language if the data for the closest language is not enough.

Source Language	Phones	Tones	Amount (hours)	Speakers	Wide-band
Cantonese	37	6	72	952	no
Vietnamese	68	6	88	954	no
Tagalog	48	n/a	85	966	no
Pashto	44	n/a	78	959	no
Turkish	42	n/a	79	990	no
Bengali	53	n/a	62	720	no
Assamese	50	n/a	61	720	no
Zulu	47	n/a	62	718	yes
Haitian	32	n/a	67	724	yes
Tamil	34	n/a	69	724	yes
Lao	43	6	66	733	yes

Table 1. Source languages data. Wideband indicates whether the language contains some amount of wideband recordings.

Target Language	Cebuano	Telugu	Swahili
VLLP training data			
Graphemes	27	57	28
Amount	3	3	3
Wideband	yes	yes	yes
Vocab	3.7k	7.3k	5.4k
Speakers	358	362	371
Web data amount	38.0M	6.4M	16.2M
Testing data			
Amount	10	10	11
Speakers	120	120	120
OOV rate	10.3	22.7	15.66
OOV rate (+web)	5.6	14.1	7.67
IV Keywords (+web)	1698	1469	1954

Table 2. Target languages used in this work. A keyword can be multiple words. A keyword is considered to be In-Vocabulary (IV), if the all the words in the keyword are IV. +web indicates values for when web data are included.

in the first two years of the program as source languages, while the languages in the third year will be the target languages. Some languages also contain a mixture of microphone data recorded at 48kHz in both train and test utterances. For the purpose of this paper, we downsampled all the wideband data to 8kHz and treated it the same way as the rest of the recordings. For the target languages, we will focus on the Very Limited Language Pack (VLLP) condition which includes only 3 hours of transcribed training data. This condition also excludes any use of human generated pronunciation dictionary. However, usage of web data is permitted for language modeling and vocabulary expansion. The list of languages and their properties can be found in Tables 1 and 2.

4.2. LID DNN analysis

We start by analyzing the effectiveness of the LID DNN in identifying the closest language. Fig. 2 shows a heat map of the averaged posteriors for each dev set speaker in Cebuano generated by the LID DNN. As shown by the figure, for the majority of the speakers, Tagalog yields the highest posterior score. This makes sense because both the Cebuano and Tagalog corpora were recorded in the Philippines. Linguistically and acoustically (channel effects) they should be the most similar. However, we also notice that the wideband recordings from Cebuano prefers languages that also include wideband recordings, while the languages without wideband recordings get little to no posterior values. This is clear evidence that the LID DNN also takes into account the acoustics as well as the linguistics, which can be more preferable than just selecting the closest language based on linguistic knowledge. This heat map also points out the need for selecting just a portion of the data from a language, since the scores can vary greatly due to different recording conditions.

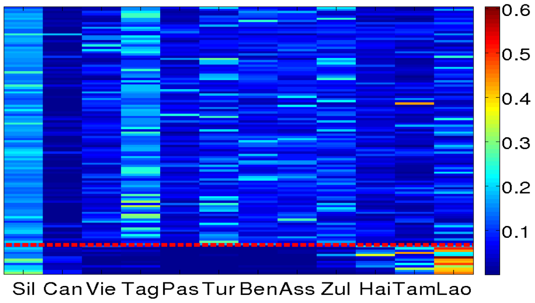

Fig. 2. A heat map of the averaged posterior scores for each speaker from Cebuano. Each row in the figure refers to a speaker. Each column refers to the language output class. The speakers below the red dashed line are from wideband recordings.

We then evaluate the averaged posteriors for each target language to identify the closest language, which we summarize in Fig. 3. To avoid the bias generated by the wideband recordings, we only use the narrowband portion to compute the average. Cebuano identifies Tagalog as the closest language followed by Lao. The top three for Telugu are Tamil, Assamese, and Bengali which are all Indian languages. Lastly, Swahili prefers Zulu. Thus, the LID DNN was able to identify the linguistically appropriate languages without any human knowledge.

4.3. Frame selection DNN analysis

We then analyze the posteriors generated by the frame selection DNNs. Fig. 4 shows the posterior values averaged over all frames for each source-target pair. The overall rankings from the LID DNN and the frame selection DNN are similar. When Telugu is the target language, Assamese, Bengali, and Tamil still remain noticeably higher than the rest of the source languages. The highest match for Cebuano is now Assamese, but Tagalog follows closely behind. Lastly, Zulu is still favored by Swahili. However, the scores are lower compared to the other two target languages. This indicates that the source languages might not be as helpful for Swahili. We also would like to note that frames with phonemes that exist in the target language tends to have higher posterior values than the ones that do not.

4.4. Recognition system

For each language, we used tied-state triphone CD-HMMs, with 2500 states and 18 Gaussian components per state. For the target languages we used a grapheme-based dictionary. Note that for IARPA-Babel languages, the difference between phonetic and graphemic systems in WER are often less than 1% [24, 25]. All the output targets of the SBN DNNs (including the multilingual SBN) were from CD states. Discriminative training was done on the CD-HMMs using the Minimum Bayes risk (MBR) criterion [26]. The web data was cleaned and filtered using techniques described in [27]. For language modeling, n-gram LMs were created from training transcripts and the web data. The vocabulary included words that appeared in

Method	Cebuano		Telugu		Swahili	
	WER	MTWV	WER	MTWV	WER	MTWV
Monolingual SBN	73.5		86.4		65.8	
Adapted multilingual	65.0	0.2259	78.0	0.1269	54.9	0.3983
Closest language	63.7	0.2526*	75.8	0.1682*	54.2	0.4225
100 hr closest frames	63.0	0.2513	76.0	0.1711*	52.4	0.4244*
200 hr closest frames	63.1	0.2531*	76.0	0.1756*	52.4	0.4233
All frames	63.0	0.2376*	75.8	0.1528*	52.4	0.4262*

Table 3. ASR and KWS results. For MTWV, * indicates the value is significantly different from one in the row above (5% significance).

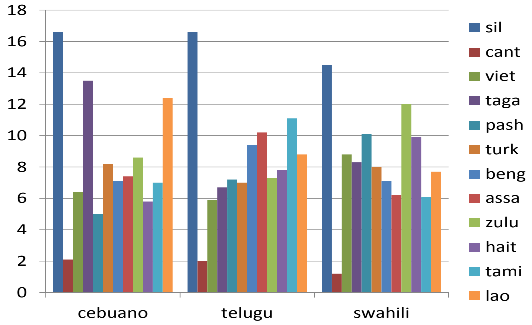

Fig. 3. LID averaged posterior scores for each target language (in percent). Only the frames from narrowband utterances are used.

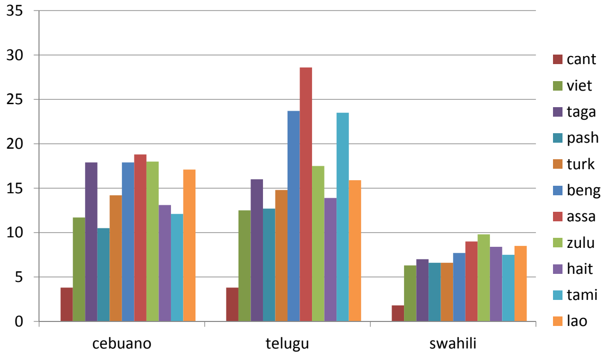

Fig. 4. Probability of being the target language averaged over all frames of each source language.

the training transcripts augmented with the top 30k most frequent words from the web.

4.5. Keyword spotting

Keyword Spotting (KWS) was done using a simplified version of what was described in [28]. For this paper, we report the KWS done using the development keywords (kwlist2) on the 10 hour dev set. We report Maximum Term-Weighted Value (MTWV) as defined in [29]. A perfect system will receive 1.0 MTWV, while a system that produces no output will receive a score of 0. We did KWS on lattices using exact word matches, since we wanted to focus more on the difference in the recognizer. For this purpose, we only report in-vocabulary (IV) keywords only.

4.6. Frame selection experiments

We compare the results between the three methods described earlier, namely a multilingual SBN adapted to the target language, a SBN trained by using the closest language as described in Section 3.1, and a SBN trained using frame selection. For frame selection, we have two configurations with 100 hours and 200 hours worth of closest frames. We did not go below 100 hours because there were too many class outputs that had no or too few frames. We also report the extreme situation where all 520 hours worth of frames are selected. Note that this is slightly different than the adapted multilingual SBN, since the second DNN for this case is trained on *adapted* BN features from the first BN. As a point of comparison, we also include a monolingual SBN trained only on the 3 hour VLLP data.

Table 3 summarizes the ASR and KWS results. Monolingual SBNs perform significantly worse than multilingual techniques. This shows the strength of using multilingual data to help ASR in languages with limited resources. Using the closest source language to train the second DNN yields a noticeable improvement over the adapted multilingual SBN in both ASR and KWS. However, the gain in WER is smaller for Swahili. This can be attributed to the fact that the candidates for Swahili are worse than the other two languages, as noted in Section 4.3. The WER differences between the closest language and closest frames methods are small. However, we note that the main goal of the Babel program, which typically operates in high WER conditions, is KWS. Minor WER differences at this level are sometimes misleading.

In terms of KWS, frame selection systems are significantly better than the ones using just the closest language. The best performance is achieved at 200 hours for Cebuano and Telugu, and 100 hours for Swahili because Swahili has lower frame selection scores. The gain from frame selection over the closest language is higher in Telugu than in Cebuano. Telugu has more than one closest language so we expect more gain from using multiple languages. Frame selection scores for the case of Telugu are also higher than Cebuano signifying better synergies between the source and target language. Finally, using all frames performs worse than any kind of selection except for the case of Swahili where the frame selection scores are noticeably lower so the effect of having more data prevails. We believe that both the amount and the closeness of the data play a role in determining the benefits from multilingual training.

5. CONCLUSION

We investigated a method to select a subset of multilingual data that would be most beneficial to train a BN feature extractor on a target language. By selecting the closest frames as scored by the frame selection DNNs, we not only were able to improve over using an adapted multilingual SBN, but also improve over our previous approach which uses just the closest language. For future work, we plan to investigate the framework for alternative setups, such as hybrid DNN-HMM, and Long-Short Term Memory networks. We also would like to look into the possibility of selecting frames from just a set of closest languages which can help decrease the training time.

6. REFERENCES

[1] L. Burget, P. Schwarz, M. Agarwal, et al., "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *Proc. ICASSP*, 2010.

[2] A. Ragni, M. Gales, and K. Knill, "A language space representation for speech recognition," in *Proc. ICASSP*, 2015.

[3] Z. Tüske, P. Golik, D. Nolden, R. Schlüter, and H. Ney, "Data augmentation, feature combination, and multilingual neural networks to improve ASR and KWS performance for low-resource languages," in *Proc. Interspeech*, 2014.

[4] E. Chuangsuwanich, Y. Zhang, and J. Glass, "Language ID-based training of multilingual stacked bottleneck features," in *Proc. InterSpeech*, 2014.

[5] F. Grézl, M. Karafiát, and K. Veselý, "Adaptation of multilingual stacked bottle-neck neural network structure for new languages," in *Proc. ICASSP*, 2014.

[6] Y. Miao, H. Zhang, and F. Metze, "Distributed learning of multilingual DNN feature extractors using GPUs," in *Proc. InterSpeech*, 2014.

[7] J. 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*, 2013.

[8] A. Mohan and R. Rose, "Multi-lingual speech recognition with low-rank multi-task deep neural networks," in *Proc. ICASSP*, 2015.

[9] F. Grézl and M. Karafiát, "Adapting multilingual neural network hierarchy to a new language," in *Proc. SLTU*, 2014.

[10] F. Grézl, E. Egorova, and M. Karafiát, "Further investigation into multilingual training and adaptation of stacked bottle-neck neural network structure," in *Proc. SLT*, 2014.

[11] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. ICASSP*, 2014.

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

[13] F. Grézl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. ASRU*, 2011.

[14] K. Veselý, M. Karafiát, F. Grézl, et al., "The language-independent bottleneck features," in *Proc. SLT*, 2012.

[15] Y. Miao, H. Zhang, and F. Metze, "Distributed learning of multilingual DNN feature extractors using GPUs," in *Proc. InterSpeech*, 2014.

[16] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, and D. Martinez, "Automatic language identification using deep neural networks," in *Proc. ICASSP*, 2014.

[17] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," in *IEEE Signal Processing Letters*, October 2015, vol. 22, pp. 1671–1675.

[18] O. Siohan, "Training data selection based on context-dependent state matching," in *Proc. ICASSP*, 2014.

[19] N. Itoh, T. Sainath, D. Jiang, J. Zhou, and B. Ramabhadran, "N-best entropy based data selection for acoustic modeling," in *Proc. ICASSP*.

[20] C. Ni, C. Leung, L. Wang, N. Chen, and B. Ma, "Unsupervised data selection and word-morph mixed language model for tamil low-resource keyword search," in *Proc. ICASSP*, 2015.

[21] K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes, "Unsupervised submodular subset selection for speech data," in *Proc. ICASSP*, 2014.

[22] T. Fraga-Silva, J. Gauvain, L. Lamel, A. Laurent, V. Le, and A. Messaoudi, "Active Learning based data selection for limited resource STT and KWS," in *Proc. InterSpeech*, 2015.

[23] *IARPA broad agency announcement IARPA-BAA-11-02*, 2011, The work uses the following language packs: Cantonese (IARPA-babel101-v0.4c), Turkish (IARPA-babel105b-v0.4), Pashto (IARPA-babel104b-v0.4aY), Tagalog (IARPA-babel106-v0.2g) and Vietnamese (IARPA-babel107b-v0.7), Assamese (IARPA-babel103b-v0.3), Lao (IARPA-babel203b-v2.1a), Bengali (IARPA-babel102b-v0.4), Zulu (IARPA-babel206b-v0.1e), Tamil (IARPA-babel204b-v1.1b), Cebuano (IARPA-babel301b-v2.0b), Telugu (IARPA-babel303b-v1.0a), and Swahili (IARPA-babel202b-v1.0d-build).

[24] V. Le, L. Lamel, A. Messaoudi, W. Hartmann, J. Gauvain, C. Woehrling, J. Despres, and A. Roy, "Developing STT and KWS systems using limited language resources," in *Proc. InterSpeech*, 2014.

[25] M.Gales, K.Knill, and A.Ragni, "Unicode-based graphemic systems for limited resources languages," in *Proc. ICASSP*, 2015.

[26] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. InterSpeech*, 2006, pp. 2406–2409.

[27] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *Proc. InterSpeech*, 2015.

[28] H. Lee, Y. Zhang, E. Chuangsuwanich, and J. Glass, "Graph-based re-ranking using acoustic feature similarity between search results for spoken term detection on low-resource languages," in *Proc. InterSpeech*, 2014.

[29] *http://www.itl.nist.gov/iad/mig/tests/std/2006/index.html*.