HIGH-PERFORMANCE SWAHILI KEYWORD SEARCH WITH VERY LIMITED LANGUAGE PACK: THE THUEE SYSTEM FOR THE OPENKWS15 EVALUATION

Meng Cai, Zhiqiang Lv, Cheng Lu, Jian Kang, Like Hui, Zhuo Zhang and Jia Liu

Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
cai-m10@mails.tsinghua.edu.cn

ABSTRACT

This paper presents the Swahili keyword search system developed by the THUEE team for the OpenKWS15 evaluation, which is conducted by NIST under the IARPA Babel program. There are several highlights in the development of the system, including automatic generation of the pronunciation lexicon, aggressive data augmentation, the multilingual bottleneck feature extractor trained from 6 languages, text selection from web data for language model training, semi-supervised training for acoustic models and language models, out-of-vocabulary keyword detection using morphemes and a rich diversity of the systems for combination. A wide variety of acoustic modeling techniques are explored and compared. Up to 12 different individual systems are used for combination. The system achieves the state-of-the-art performance in the required condition of the evaluation.

Index Terms— Speech recognition, keyword search, deep neural network (DNN), low-resource, acoustic model

1. INTRODUCTION

Ever since 2013, the National Institute of Standards and Technology (NIST) has conducted a series of keyword search evaluations called OpenKWS under the IARPA Babel program [1]. These evaluations aim at developing automatic speech recognition (ASR) systems for the keyword search tasks using limited data resources in a short period of time. A "surprise" language is released each time, whose identity is unknown before the evaluation. The surprise languages are Vietnamese for the OpenKWS13 and Tamil for the OpenKWS14. The OpenKWS evaluations are open to the Babel performers as well as the volunteers, providing benchmark results for the ASR community.

The OpenKWS15 [2] is the most recent evaluation in this series. The surprise language is Swahili this time. Compared with the previous OpenKWS13 and OpenKWS14 evaluations, several new rules make the OpenKWS15 evaluation much more challenging. First, the required condition for the volunteers is the very limited language pack (VLLP) condition, in which there is only 3 hours of transcribed training data plus 40 hours of untranscribed training data. The required full language pack (FullLP) conditions in previous evaluations, however, typically contain 60 to 80 hours of transcribed data. Second, the participants are provided with several language packs other than Swahili, making it possible to deploy multilingual techniques. Third, the pronunciation lexicon is not included in the language pack. The participants have to derive the lexicon and the phoneme set. Fourth, a large amount of web data is provided along with the VLLP data for training the language models. Fifth, the system build time is reduced from 3 to 2 weeks. These evaluation rules are useful for rapidly building practical keyword search systems for new languages.

Existing methods for keyword search mainly fall into two categories. One is the query-by-example (QbE) based keyword search [3, 4]. The other is the large vocabulary continuous speech recognition (LVCSR) based keyword search [5]. The QbE based methods are fast and language independent. But they are not suitable for searching a large number of keywords because the examples of every keyword are needed as the templates. The LVCSR based methods, on the other hand, are scalable to deal with thousands of keywords, as is the case in the OpenKWS15 evaluation. In the LVCSR based methods, the acoustic models (AMs) and the language models (LMs) are trained, then rich lattices are generated for keyword search. The LVCSR based methods have been proved effective in previous NIST spoken term detection evaluations [6]. But LVCSR methods typically rely on large amounts of training data of a specific language. Dealing with the problem of low data resource is a major concern in deploying LVCSR based methods for the OpenKWS15 evaluation task.

In this paper, we introduce the THUEE team's methods for the OpenKWS15 evaluation, which have achieved state-of-the-art results. We focus on the required VLLP condition for the volunteers. The latest techniques in LVCSR are explored for the evaluation, such as the long short-term memory (LSTM) recurrent neural network (RNN) for acoustic modeling [7], the combination of the convolutional structure and the

This work is supported by National Natural Science Foundation of China under Grant No. 61273268, No. 61370034, No. 61403224 and No. 61005017.

recurrent structure for acoustic modeling [8], the maxout networks [9] and its variants [10, 11] for acoustic modeling, neural network language models (NNLM) for first-pass decoding [12], etc. The techniques for low resource ASR are also explored, such as the multilingual bottleneck (MBN) features [13], data augmentation [14], semi-supervised acoustic model training [15], etc. These techniques are integrated and detailed experimental results are compared. We hope this work could provide insights for researchers working on the keyword search tasks for low resource languages.

2. BASELINE SETUP

2.1. Task definition

The keyword search task is to find the occurrences of a list of keywords in a corpus of speech data. The keywords can be one or more words in a language. In the OpenKWS15 evaluation, a keyword search system has to output the begin time and the duration of all the possible detected keywords in their corresponding audio files, along with the confidence scores. A global threshold θ is used by the system to make the hard decision whether a detected keyword is correct. The performance of a keyword search system is judged by the actual term-weighted value (ATWV), which is defined as

$$\mathrm{ATWV}(\theta) = 1 - \frac{1}{K}\sum_{w=1}^{K}\left(\frac{\#miss(w,\theta)}{\#ref(w)} + \beta\frac{\#fa(w,\theta)}{T - \#ref(w)}\right), \quad (1)$$

where $\#ref(w)$ is the number of reference occurrences of the keyword w, $\#miss(w,\theta)$ and $\#fa(w,\theta)$ are the number of missed detections and the number of false alarms of the keyword w at threshold θ. T is the total duration of the corpus in seconds, which is an approximation of the number of trials. K is the total number of different keywords and β is a constant set at 999.9. The optimal threshold θ results in the maximum term-weighted value (MTWV). In addition to the ATWV, the word error rate (WER) of the system is also required. So we report both the AWTV and the WER in the experiments.

2.2. Data description

The OpenKWS15 language pack includes a training set, a tuning set, a development set and an evaluation set in the VLLP condition. The training set consists of 3 hours of transcribed conversational telephone speech data and 40 hours of untranscribed telephone speech data for model training. The tuning set consists of 3 hours of transcribed data for parameter tuning. The development set consists of 10 hours of transcribed data for evaluation by the participants themselves. The evaluation set consists of about 75 hours of data for the final evaluation by NIST. There is also a collection of text data scraped from the web (about 270 megabytes in gzip compressed format) for enhancing the language models. The pronunciation lexicon is not available in the language pack. But a language specific peculiarities (LSP) document is provided, which contains the letter-to-sound rules for Swahili.

Table 1. The release ID and the size (hours) of transcribed speech data in the training corpus of the languages.

Language	Release ID	Size
Cantonese	IARPA-babel101b-v0.4c	140.7
Pashto	IARPA-babel104b-v0.4bY	77.3
Turkish	IARPA-babel105b-v0.4	76.3
Tagalog	IARPA-babel106-v0.2g	83.7
Vietnamese	IARPA-babel107b-v0.7	87.1
Tamil	IARPA-babel204b-v1.1b	62.3
Swahili	IARPA-babel202b-v1.0d	3.1

In addition to the Swahili language resource, the participants are also provided the language resources of several languages under the Babel program. Each of the language consists a FullLP training set and a development set of about 10 hours. The volunteers are allowed to use 6 Babel languages, as shown in Table 1.

There are two versions of keyword lists provided by NIST. The first version is generated by IBM and BBN during system development, which contains 2480 keywords. The second version is the official evaluation keyword list released during system evaluation, which contains 4454 keywords. For consistency we report the results for the first version of the keyword list on the development set[1].

2.3. Generation of the pronunciation lexicon

The very first step in building an LVCSR system for the OpenKWS15 evaluations is to generate the pronunciation lexicon. According to previous study, the pronunciation of a Swahili word can be derived from its spelling [16]. We generate the grapheme-to-phoneme mapping using the letter-to-sound rules from the LSP document. A total of 40 non-silence phonemes are used, including 33 phonemes for consonants, 5 phonemes for vowels and 2 phonemes for the voice of hesitations. A phoneme with 4 hidden Markov model (HMM) states is used to model the silence frames.

As there is only 3 hours of transcribed data in the VLLP training set, the vocabulary size for the VLLP training set is only 5357. However, Swahili is a morpheme-rich language. Using a vocabulary of 5357 words should result in very high out-of-vocabulary (OOV) rate. We use three methods to deal with the OOV problem. First, we use the words in the web text to enrich the vocabulary size. More specifically, about 100,000 most frequent words from the web text are selected and added to the vocabulary. The actual vocabulary size for our system is 109,966. Second, the method of applying

[1] At the time of the paper the results on the evaluation set are unreleasable beyond the OpenKWS teams.

Table 2. The WERs (%) of the baseline systems and the systems with data argumentation. The systems are trained with PLP plus pitch features. VTLP: vocal tract length perturbation. SGMM: subspace Gaussian mixture model. BMMI: boosted maximum mutual information.

System	Baseline		Noising		VTLP		Noising+VTLP	
	tuning	dev	tuning	dev	tuning	dev	tuning	dev
GMM	66.3	65.2	66.5	65.8	66.0	64.9	65.1	64.0
GMM BMMI	68.3	66.4	67.6	64.7	66.6	64.3	65.2	62.8
SGMM	61.7	61.6	61.9	62.5	61.6	61.0	60.4	60.1
SGMM BMMI	62.8	62.7	61.0	60.9	61.2	60.2	59.6	58.8

proxies for OOV keywords is used in keyword search [17]. Third, the morpheme language models are used to generate morpheme lattices, which are used for keyword search and system combination. The details of the morpheme language models are presented in Subsection 3.5.

2.4. Baseline results

The baseline system is built with the 3 hours of transcribed VLLP data using the *Kaldi* toolkit [18]. The 13-dimensional PLP features plus 3-dimensional pitch features [19] are first extracted. Then 9 consecutive feature frames are concatenated and the feature dimension is reduced to 40 using linear discriminant analysis (LDA) plus a global semi-tied covariance transform. The 40-dimensional features are used to train a Gaussian mixture model-hidden Markov model (GMM-HMM), which contains about 2000 states and 15000 Gaussian mixtures. The GMM-HMM is first trained using feature-space maximum likelihood linear regression (fMLLR) based speaker adaptive training (SAT) and then enhanced using the boosted maximum mutual information (BMMI) criterion. A subspace GMM (SGMM) based acoustic model [20] is also trained, which contains about 2000 states, 120 Gaussian mixtures per state and 8000 substates. We test the WERs on the tuning and the development set, with the language model trained using the transcriptions of the VLLP training data. The results are shown in the baseline columns of Table 2.

The baseline results show that the SGMM-HMM models have relative improvements of between 5.5% to 8.1% over the GMM-HMM models, which confirms the SGMM-HMM model's better performance in low resource acoustic modeling than the GMM-HMM models. However, the BMMI discriminative training degrades the performance due to the lack of training data. These results inspire us to explore specific methods for the VLLP condition to improve the models, especially the models after discriminative training.

3. SYSTEM IMPROVEMENTS

In this section we discuss and compare the improvements we explored for the systems.

3.1. Data augmentation

To compensate for the lack of training data, the most direct way is to "add data". Using transforms of the data as the input while preserving the labels has long been proved effective for neural network training. In this work we explore two data argumentation methods for the OpenKWS15 evaluation, namely adding noise and vocal tract length perturbation (VTLP) [21].

We try four kinds of noise types in the experiments. These noise types include babble noise, pink noise, subway noise and white noise. For each of the noise type, the signal noise ratio (SNR) of 20, 25 and 35 are tried. So a total of 36 hours of noisy data is generated. We randomly choose 9 hours of data from the generated data set and add it to the original 3-hour training set, forming the noising training set. The GMM-HMM and the SGMM-HMM acoustic models are trained in the same way as in the previous experiments. The results are shown in the noising columns of Table 2.

For the VTLP experiments, the warp factor of 0.92, 0.96, 1.04 and 1.08 are tried. The generated data is added to the original 3-hour training set, forming the VTLP training set. The results of the VTLP data augmentation is shown in the VTLP columns of Table 2.

We also try to mix the noising training set and VTLP training set together. In this way a training set of 24 hours of augmented data is formed. The results are shown in the last two columns of Table 2.

From the data augmentation results, we see that both the noising method and the VTLP method can improve the performance of the baseline. The VTLP method yields superior performance than the noising method. A combination of both the noising method and the VTLP method produces even better results. An interesting phenomenon is that the data augmentation methods are particularly effective for the models with sequence training. Even though the sequence training degrades the performance in the baseline results, it is mostly helpful in conjunction with the data augmentation methods. The reason for this phenomenon is that the augmented training data produces richer denominator lattices during discriminative training, which reduces overfitting for the discriminative training.

Table 3. The WERs (%) of the systems trained with multilingual bottleneck (MBN) features. AUG: data augmentation, using both noising and VTLP.

System	MBN		MBN+AUG	
	tuning	dev	tuning	dev
GMM	62.9	61.2	63.0	63.0
GMM BMMI	61.6	59.2	60.9	58.9
SGMM	55.8	53.0	55.2	53.1
SGMM BMMI	55.9	52.9	53.4	51.4

3.2. Multilingual bottleneck features

Apart from the data augmentation method, another way to compensate for the lack for training data is to "borrow data". In this work we borrow data from the 6 Babel languages using multilingual bottleneck (MBN) features.

We train a DNN as the MBN feature extractor using the 6 Babel languages. The inputs of the DNN are 40-dimensional Mel filterbank features plus 3 dimensional pitch features, together with their first- and second-order derivatives. A context window of 11 frames is applied. The DNN has 7 hidden layers. The sixth hidden layer (counting from the input layer) is a linear layer with 128 neuron, which produces the MBN features. Other hidden layers are sigmoid layers with 1500 neurons. Instead of using the multitask learning methods [22], we pool all the phonemes of the languages together and generate 7519 context-dependent triphone states, which is similar to the method in [13]. The DNN is trained using stochastic gradient descent (SGD) to optimize the cross entropy (CE) target.

After the 128-dimensional MBN features are extracted, the 9 consecutive feature frames are concatenated and the feature dimension is reduced to 40 using LDA and semi-tied covariance transform. The GMM-HMMs and SGMM-HMMs are trained in the same way as in previous experiments. The results are given in Table 3. The models in the MBN columns are trained using the 3 hours of VLLP data. The models in the MBN+AUG columns are trained with the 24 hours of augmented data using the method in the previous subsection. Comparing with the results in Table 2, we see that there are relatively large gains brought by the MBN features. The relative improvements of the SGMM-HMM models trained using the augmented MBN features are between 10.4% to 12.6% compared with the SGMM-HMM trained using the augmented PLP plus pitch features. It is also shown in Table 3 that the gain brought by the data augmentation method preserves for MBN features.

3.3. Text selection from web data

The NIST provides a collection of web text data. However, this text data set is very "noisy", e.g., there are lots of web URLs, English words and punctuation marks. Moreover, the contents of the web text are quite different from the contents of telephone conversations. Thus applying the web text is nontrivial for the language model training.

To make good use of the web text data, we use the data selection method based on the difference of cross entropy scores between in-domain and out-of-domain text. The transcriptions of the VLLP training data serve as the in-domain data and the transcriptions of the tuning set are used for development. An open source toolkit, *XenC* [23], is used for the data selection. The selected text is used to train a trigram language model. Then an interpolation is conducted with the original language model trained from the VLLP training set transcriptions. We find that selecting 1 million sentences from the web text is the most helpful strategy to optimize the perplexity on the tuning set. The WERs as well as the perplexities of the LMs are shown in the first two rows of Table 4.

Table 4. The WERs (%) of the systems with different data for training the acoustic models (AMs) and the language models (LMs) as well as the perplexities (PPLs) of the LMs. The semi-supervised data is decoded with a DNN-HMM model that achieves a WER of 49.2% on the development set.

GMM, MBN+AUG	PPL	WER	
	tuning	tuning	dev
baseline	937.0	63.0	63.0
LM+web	775.6	62.4	62.2
AM+semi, LM+web	775.6	61.7	59.8
AM+semi, LM+web+semi	672.7	61.1	59.5

3.4. Semi-supervised training

The methods to "add data" and "borrow data" are both effective in previous experiments. In this experiment we try to "mine data", i.e., perform semi-supervised training for both the acoustic models and the language models.

To perform semi-supervised training, the 40-hour untranscribed training data needs to be decoded. We train a DNN-HMM acoustic model [24, 25, 26] with the MBN features and the 24 hours of augmented data set to decode the untranscribed training data. The DNN contains 6 hidden layers and 1000 sigmoid neurons per hidden layer. The input dimension is 1408 (128×11) and the output dimension is 2006. The SGMM-HMM model with MBN features and augmented data is used to generate the alignments for DNN training. The DNN is first trained by the CE criterion and then refined by state-level minimum Bayes risk (sMBR) sequence training [27, 28]. The language model enhanced with web data is used. This DNN based system achieves a WER of 49.2% on the development set.

We use all the untranscribed data with the decoded transcriptions for the semi-supervised training. We first try to add the semi-supervised data to train the acoustic models. The a-

Table 5. The WERs (%) of the systems with MBN features and data augmentation methods. Semi-supervised data and web text are used to enhance the models.

System	tuning	dev
GMM	61.1	59.5
GMM BMMI	58.9	55.0
SGMM	54.1	51.1
SGMM BMMI	52.7	49.7

coustic models with semi-supervised data contain about 5000 tied-triphone states. The results are shown on the third row of Table 4. Then we train a language model using the decoded text and interpolate it with the original language model and the web text language model. The results of using semi-supervised data for both the acoustic model and the language model are shown in the fourth row of Table 4. The results show that the semi-supervised data are helpful for both the acoustic model training and the language model training.

Table 5 shows the results of the systems with semi-supervised training for the AMs and the LMs. The results indicate that the semi-supervised data, though not accurate, is still helpful for discriminative training.

3.5. Morphemes for OOV keywords

Swahili is an agglutinative language with a huge vocabulary size. Even though we use a lexicon of more than 100,000 word items, the OOV rate is still high. The morpheme based methods have been proved effective for speech recognition of agglutinative languages [29], as the words in agglutinative languages are consisted of a fixed set of morpheme units. In this work we also adopt the morpheme based methods for keyword search.

The morphemes for Swahili are discovered in an unsupervised fashion using the *Morfessor* toolkit [30]. A total of 11,362 morpheme units are automatically generated. Then the trigram morpheme LM is obtained by interpolating the morpheme LMs trained from the VLLP transcriptions, the selected web text and the decoded untranscribed data. The morpheme lattices are generated with every individual system. Finally the keyword search results from the word lattices and the morpheme lattices are combined in the experiments.

4. OVERALL PERFORMANCE

In this section we show the overall performance of the final systems. The performances of the individual systems are first given. The system combination results are then presented.

4.1. Individual systems

In pursuit of high-performance keyword search, we make our best effort in terms of acoustic modeling and system diversity.

Table 6. The performance of the final individual systems on the development set. System S1, S2, S3 and S12 are trained with CE criterion. System S4–S10 are trained with sMBR criterion. System S11 is trained with BMMI criterion. System S1–S11 are built upon Kaldi. System S12 is built upon HTK.

System	WER	MTWV	
		word	morph
S1: maxout, MBN	47.8	0.4794	0.3477
S2: p-norm, MBN	47.3	0.4712	0.3557
S3: CMNN, fbank	49.4	**0.4829**	**0.3570**
S4: DNN, PLP+pitch	51.9	0.4562	—
S5: DNN, fbank+pitch	49.4	0.4675	0.3446
S6: DNN, MBN	47.3	0.4666	0.3503
S7: DNN, SAT	48.4	0.4712	0.3514
S8: RNN, MBN	**47.2**	0.4778	0.3445
S9: LSTM, MBN	48.5	0.4333	0.3180
S10: CRNN, fbank	51.0	0.4419	0.3228
S11: SGMM, MBN	49.7	0.4331	0.3135
S12: DNN, SAT, NNLM	53.3	0.4308	0.3558

A total of 12 individual systems are built. All of the acoustic models are trained with the 24 hours of augmented data plus the 40 hours of semi-supervised data. Approximately 5000 context-dependent triphone states are used for all the acoustic models. A word trigram LM and a morpheme trigram LM are used for each individual system, which are trained using the transcriptions of the VLLP training data, the selected web text and the decoded transcriptions of the untranscribed data as mentioned in the previous section. Totally 12 sets of word lattices and 11 sets of morpheme lattices are generated for keyword search. The overall performance of the individual systems are presented in Table 6. The MTWVs are obtained after the keyword-specific thresholding and exponential (KST) normalization [31].

System S1 is based on the deep maxout network AM [32] with the MBN features. The network has 7 hidden layers and 1000 maxout neurons per hidden layer. Each maxout neuron contains 2 alternative pieces. The network is trained by SGD regularized by the dropout strategy [33]. A dropout rate of 0.2 is used for all the hidden layers.

System S2 is based on the p-norm maxout neural network AM [11] with the MBN features. The network contains 4 hidden layers and 300 p-norm neurons per hidden layer. We use $p = 2$ and a group size of 10 for each p-norm neuron, as suggested in [11]. Training is performed using the parallel version of SGD based on natural gradient and parameter averaging [34].

System S3 is based on the convolutional maxout neural network (CMNN) AM [10] with 40-dimensional Mel filterbank features and their first- and second-order derivatives. Two convolutional layers with 256 maxout neurons and five

fully-connected layers with 1000 maxout neurons are used. Each maxout neurons contains 2 pieces. A max-pooling layer with a pooling size of 3 is used between the convolutional layers. The first convolutional layer has a band width of 8. The second convolutional layer has a band width of 4. Dropout training is applied to the fully-connected part of the CMNN.

System S4, S5, S6 and S7 are based on fully-connected DNN AMs trained with different features. The feature types include the PLP plus pitch features, the Mel filterbank plus pitch features, the MBN features and the fMLLR speaker adapted MBN features. These AMs are first trained with the CE criterion and then refined using the sMBR criterion.

System S8 is based on an RNN AM [35] with MBN features. The first hidden layer is recurrent with 1500 neurons, followed by 4 fully connected layers with 1500 neurons.

System S9 is based on the LSTM RNN AM with sMBR sequence training [7, 36]. Three LSTM layers with 1024 neurons are used. A linear projection layer of 512 dimensions follows each LSTM layer. The truncated backpropagation through time (BPTT) algorithm with a subsequence length of 20 is used to train the model. The input to the LSTM RNN is a single frame of MBN feature, while the target label is delayed for 5 frames.

System S10 is based on the convolutional recurrent neural network (CRNN) AM. A convolutional layer is first applied to the Mel filterbank features, followed by the pooling layer, followed by another convolutional layer. Then a recurrent layer is applied with 1500 neurons, followed by 2 fully-connected layers with 1500 neurons and a final softmax output layer.

System S11 is based on the SGMM AM [20], which contains 600 Gaussian mixtures per state and 20000 substates.

System S12 is built upon the *HTK* toolkit [37]. A DNN AM is trained using the same feature type as in System S7. The DNN is trained with the CE criterion. We also train a word NNLM and a morpheme NNLM with the variance regularization for one-pass lattice generation [12]. The context lengths of the NNLMs are set to 4. The dimensions of the word features and the hidden layers are set to 300.

The results in Table 6 reveal several interesting phenomena. First, systems having better WER results may not necessarily have better keyword search results. Second, the MBN features are more effective to optimize the WER, but less effective for keyword search. We believe the reason for the two phenomena is that the AMs that produce better WERs may have sharp distributions for their output posteriors (e.g. the RNNs or the models with MBN features), which reduces the lattice sizes and degrades the keyword search performances.

4.2. System combination

The system combination makes the best use of the system diversities, which is a crucial part for the OpenKWS evaluations [38, 31, 39, 40]. We use the WCombMNZ method proposed in [38]. The optimal strategies are obtained by brute-force search of all the possible combinations of the systems, including the word based systems and morpheme based systems. The original confidence scores are used before combination and KST normalization is applied after combination. The results of our primary system (System P1) and two contrast systems (System C1 and C2) are shown in Table 7. For the System P1, we use 11 word based systems (System S2–S12) and 3 morpheme based systems (System S1, S6 and S7) for in-vocabulary (IV) search. The same 11 word based systems and 9 morpheme based systems (System S1, S2, and S5–S11) are used for OOV search. System C1 uses 11 word based systems and 9 morpheme based systems (same as the OOV search for System P1) to search all the keywords. System C2 uses 11 word based systems and 3 morpheme based systems (same as the IV search for System P1) to search all the keywords. During the system combination process, we empirically find that using a large number of diversified systems is beneficial for the keyword search performance [41]. Moreover, the optimal thresholds for the MTWVs tend to be stable when a large number of systems are combined.

Table 7. The ATWVs and MTWVs of the system combinations on the development set. The thresholds for the ATWVs are determined on the tuning set.

System	ATWV	MTWV
C1: 11 word + 9 morph	0.5670	0.5670
C2: 11 word + 3 morph	0.5715	**0.5722**
P1: IV 11 word + 3 morph, OOV 11 word + 9 morph	**0.5717**	0.5721

For the speech-to-text (STT) task, we use the ROVER method [42] to combine the output of all the word based systems. The combined system achieves a WER of 43.5% on the development set. Excluding any of the individual systems results in degradation of the WER performance. The ATWV of **0.5717** and the WER of **43.5%** are state-of-the-art results for the OpenKWS15 evaluation among the participating teams of volunteers under the VLLP condition.

5. CONCLUSIONS

In this paper, we have proposed a state-of-the-art Swahili keyword search system for the OpenKWS15 evaluation using the very limited language pack. It is found that data augmentation methods are effective for the models after discriminative training. The multilingual bottleneck features produce relative improvements of more than 10% over the PLP features for the SGMM-HMM system. Adding selected web text data and semi-supervised data are both helpful for system performance. Exploring a broad class of acoustic modeling techniques, we find that systems with better WERs may not have better ATWVs. Combining a large number of systems results in good performances and stable thresholds for the ATWVs.

6. REFERENCES

[1] IARPA, "The Babel Program," http://www.iarpa.gov/index.php/research-programs/babel.

[2] NIST, "KWS15 keyword search evaluation plan," http://www.nist.gov/itl/iad/mig/upload/KWS15-evalplan-v05.pdf, 2015.

[3] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 38, no. 11, pp. 1870–1878, Nov. 1990.

[4] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*. Florence, Italy, 2014, pp. 4087–4091.

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

[6] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. ACM SIGIR Workshop on Searching Spontaneous Conversational*, 2007, pp. 51–57.

[7] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*. Singapore, 2014, pp. 338–342.

[8] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. ICASSP*. Brisbane, Australia, 2015, pp. 4580–4584.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. ICML*. Atlanta, USA, 2013.

[10] M. Cai, Y. Shi, J. Kang, J. Liu, and T. Su, "Convolutional maxout neural networks for low-resource speech recognition," in *Proc. ISCSLP*. Singapore, 2014, pp. 133–137.

[11] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*. Florence, Italy, 2014, pp. 215–219.

[12] Y. Shi, W.-Q. Zhang, M. Cai, and J. Liu, "Efficient one-pass decoding with NNLM for speech recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 377–381, Apr. 2014.

[13] K. M. Knill, M. J. F. Gales, S. P. Rath, P. C. Woodland, C. Zhang, and S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Proc. ASRU*. Olomouc, Czech Republic, 2013, pp. 138–143.

[14] X. Cui, V. Voel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 9, pp. 1469–1477, Sep. 2015.

[15] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Proc. ASRU*. Olomouc, Czech Republic, 2013, pp. 267–272.

[16] H. Gelas, L. Besacier, and F. Pellegrino, "Developments of Swahili resources for an automatic speech recognition system," in *Proc. SLTU*, 2012.

[17] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. ASRU*. Olomouc, Czech Republic, 2013, pp. 416–421.

[18] 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. ASRU*. Hawaii, USA, 2011.

[19] P. Ghahremani, B. BabaAli, and D. Povey, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proc. ICASSP*. Florence, Italy, 2014, pp. 2513–2517.

[20] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiat, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model—a structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, pp. 404–439, 2011.

[21] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *Proc. ICML*. Atlanta, USA, 2013.

[22] 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*. Vancouver, Canada, 2013, pp. 7304–7308.

[23] A. Rousseau, "XenC: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, pp. 73–82, 2013.

[24] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*. Florence, Italy, 2011, pp. 437–440.

[25] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[26] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[27] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*. Portland, USA, 2012, pp. 10–13.

[28] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*. Lyon, France, 2013, pp. 2345–2349.

[29] M. Ablimit, T. Kawahara, and A. Hamdulla, "Lexicon optimization based on disciminative learning for automatic speech recognition of agglutinative language," *Speech Commun.*, vol. 60, pp. 78–87, 2014.

[30] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech, Lang. Process.*, vol. 4, no. 1, Jan. 2007.

[31] 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. Le, "Score normalization and system combination for improved keyword spotting," in *Proc. ASRU*. Olomouc, Czech Republic, 2013, pp. 210–215.

[32] M. Cai, Y. Shi, and J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*. Olomouc, Czech Republic, 2013, pp. 291–296.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[34] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of DNNs with natural gradient and parameter averaging," *arXiv:1410.7455*, 2015.

[35] G. Saon, H. Soltau, A. Emami, and M. Picheny, "Unfolded recurrent neural networks for speech recognition," in *Proc. Interspeech*. Singapore, 2014, pp. 343–347.

[36] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Proc. Interspeech*. Singapore, 2014, pp. 1209–1213.

[37] S. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK book, version 3.4.1*, Cambridge University Engineering Department, 2009.

[38] 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. ICASSP*. Vancouver, Canada, 2013, pp. 8272–8276.

[39] V. T. Pham, N. F. Chen, S. Sivadas, H. Xu, I.-F. Chen, C. Ni, E. S. Chng, and H. Li, "System and keyword dependent fusion for spoken term detection," in *Proc. SLT*. South Lake Tahoe, USA, 2014, pp. 430–435.

[40] M. Cai, Z. Lv, B. Song, Y. Shi, W. Wu, C. Lu, W.-Q. Zhang, and J. Liu, "The THUEE system for the OpenKWS14 keyword search evaluation," in *Proc. ICASSP*. Brisbane, Australia, 2015, pp. 4734–4738.

[41] Z. Lv, M. Cai, C. Lu, J. Kang, L. Hui, W.-Q. Zhang, and J. Liu, "Improved system fusion for keyword search," in *Proc. ASRU*. Scottsdale, USA, 2015.

[42] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. ASRU*. Santa Barbara, USA, 1997, pp. 347–354.