THE IBM KEYWORD SEARCH SYSTEM FOR THE DARPA RATS PROGRAM

Lidia Mangu, Hagen Soltau, Hong-Kwang Kuo and George Saon

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

ABSTRACT

The paper describes a state-of-the-art keyword search (KWS) system in which significant improvements are obtained by using Convolutional Neural Network acoustic models, a two-step speech segmentation approach and a simplified ASR architecture optimized for KWS. The system described in this paper had the best performance in the 2013 DARPA RATS evaluation for both Levantine and Farsi.

Index Terms— spoken term detection, keyword spotting, audio indexing, system combination

1. INTRODUCTION

One of the fundamental problems in automatic speech processing is finding a spoken or written `keyword` in a collection of audio recordings. By `keyword` we mean one word or a sequence of words. Research in keyword search, also known as spoken term detection (STD), has been substantially advanced by competitive evaluations. The first evaluation was the NIST STD 2006 evaluation [1], in which participants built systems for English, Arabic and Chinese. Recently, there is renewed interest in evaluating spoken term detection systems in a variety of languages and audio conditions. In the DARPA RATS (Robust Automatic Transcription of speech) program, existing Arabic Levantine and Farsi telephone conversations are retransmitted through 8 different communication channels with different degrees of noise and channel distortion. One of the four RATS tasks is keyword search on this highly degraded speech. In this paper we describe the system we deployed in the second RATS evaluation, which was held in February 2013. We use a pre-indexed system in which the audio to be searched is indexed without prior knowledge of the query terms. This approach is beneficial when large amounts of audio are to be searched interactively. The same index and the same query expansion is used for both in-vocabulary (IV) and out-of-vocabulary (OOV) queries; the only additional work required for OOVs is the generation of pronunciations.

In Section 2 we describe the evaluation data and metrics. In Section 3 we present various audio segmentation methods that were explored for this task. The next two sections show the acoustic models and ASR architecture used to produce lattices needed for indexing. Section 6 gives an overview of our WFST-based indexing and search system. In Section 7 we describe the system used in the 2013 DARPA RATS evaluation and we conclude in Section 8.

2. DATA AND METRICS

All the training and test sets for the DARPA RATS program are provided by the LDC (Linguistic Data Consortium) [2, 3]. The languages used in the second phase of the DARPA RATS KWS evaluation are Arabic Levantine and Farsi. Much of the Levantine clean speech data is existing Fisher data that was re-purposed for the RATS program. To introduce signal degradation, the clean speech was transmitted through eight different radio channels, labeled A–H, corresponding to different transmitter/receiver pairs, and then recorded. The clean data contained about 150h of audio, but only about 65h was labeled as speech. In the end, for acoustic model training, we defined a 251h noisy speech (N) training set and a 310h noisy plus clean (N+C) set. To train the language model, we used only the transcripts corresponding to the 65h of clean speech (about 500K words) because the transcripts for the other channels are exactly the same.

The Farsi data was similar to the Levantine data in terms of how the clean data was degraded to produce the noisy data. In the end, we ended up with about 40h of labeled clean speech audio and 300h of noisy plus clean speech, which we use for training the acoustic models. The language model was trained on about 500K words.

The probability of miss (pMiss) is defined to be $\frac{\sum_i \#\text{times keyword } i \text{ is missed}}{\sum_i \#\text{occurrences of keyword } i}$. The probability of false alarm (pFA) is defined to be $\frac{\#\text{false alarms}}{\#\text{total words} \times \#\text{keywords}}$. In the Phase 2 DARPA RATS evaluation, the goal is to minimize pFA at an operating point of 20% pMiss; this metric is denoted pFA@20%pMiss.

For Levantine we will report results on two sets `dev-1` and `dev-2` . `dev-1` is the development data consisting of 219 keywords to search in 2.4 h of audio. `dev-2` is the Phase 1 evaluation test data and consists of 200 words to search in 34.2 h of audio.

For Farsi, we have only one development set consisting of 200 words to search in 17 h of audio. For the official DARPA evaluation, there is also a sequestered progress set for both Levantine and Farsi.

3. AUDIO SEGMENTATION

As we reported in [4], it is beneficial for the KWS task to combine systems using diverse audio segmentations. For the Phase 2 KWS evaluation system we used two segmentations, corresponding to our Phase 1 and 2 RATS Speech Activity Detection (SAD) evaluation systems.

The first audio segmentation variant (S1) is based on HMM Viterbi decoding with 3 states corresponding to speech, silence and no-transmission. It uses channel-dependent GMMs and neural networks that are trained in a 40-dimensional LDA feature space obtained by projecting consecutive PLP cepstra within a time window of $\pm$ 4 frames. Additionally, both GMMs and neural networks are estimated with boosted MMI with an asymmetric loss function that only penalizes false alarm frames. During segmentation, the scores from the GMMs and the neural networks are log-linearly combined at the frame level.

The second segmentation variant (S2) is done using a speech/non-speech HMM Viterbi decoder with channel-dependent deep neural networks (DNNs) trained on a fusion of PLP, voicing and frequency-domain linear prediction (FDLP) features. The PLP and FDLP cepstra are normalized to zero mean and unit variance using audio file-based statistics. Additionally, we apply cepstral averaging for each dimension within a temporal window of $\pm$20 frames.

To each normalized PLP frame we append a 1-dimensional voicing feature yielding a 14-dimensional frame. Every 17 consecutive PLP+voicing frames are spliced together and projected down to 40 dimensions using linear discriminant analysis (LDA). Similarly, every 17 consecutive FDLP frames are spliced together and projected down to 40 dimensions using LDA. The LDA-projected PLP and FDLP features are concatenated and augmented with single, double and triple deltas leading to input features of size $(40+40)\times 4 = 320$. The DNNs have 3 hidden layers of 1024 neurons each and sigmoid transfer functions. The output layer has 3 neurons (corresponding to speech, non-speech and non-transmission) with softmax transfer function. Training is done with minimum cross-entropy and takes around 30 passes (epochs) through the training data to converge. More details about the segmentation system can be found in [5].

Audio segmentations for the KWS task should have the following properties: (1) reduced speech miss rate, and (2) short audio segments which allow for deeper search without increasing the lattice size (search beams). In order to achieve (1) we optimize SAD parameters to decrease the miss rate, and for (2) we use a 2-step approach:

- Produce audio segments using a SAD system optimized to decrease the speech miss rate
- Decode the resulting segments and split the segments according to the position and duration of the non-speech events in the decoded output. We found that the end-of-sentence symbol was the most informative cutting point, with the added benefit of generating sentence-like segments.

Figure 1 shows that this resegmentation approach results in significantly better KWS results.

There are many ways in which we can use diverse audio segmentations for KWS. We can either combine the segmentations before ASR and KWS, i.e. we union the segments and generate a segmentation that will cover more speech. This method has the benefit of having to decode and search only once. The second method involves using the two segmentations in parallel and combining the posting lists after ASR and KWS. Figure 2 compares the two methods. This result shows that decreasing the missed speech is not the only dimension that is important for KWS.

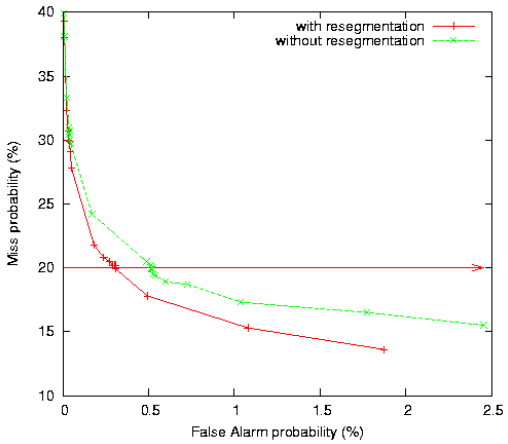

Fig. 1. 1-pass vs 2-pass audio segmentation approach

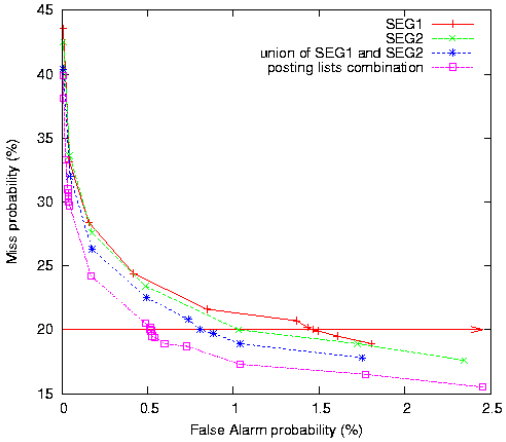

Fig. 2. Comparison of different methods for combining audio segmentations

4. NEURAL NETWORK ACOUSTIC MODELS

Our Phase 1 RATS system [4] was based on Gaussian Mixture Models and serves as a baseline for our Neural Net acoustic models. The GMM models use all the conventional LVCSR techniques including VTLN, LDA, STC, FMLLR, MLLR. The models are trained with both feature and models space boosted MMI. The baseline error rates are shown in Table 2. While the error rates seem high, one has to keep in mind that this task is rather difficult and deals with telephone conversational speech and is re-transmitted over noisy channels. Our GMM uses all of the state-of-the-art techniques, and achieved excellent performance in the Phase 1 RATS evaluation [4].

4.1. Multi-Layer Perceptrons (MLP)

While Neural Nets for Speech recognition are not new, they did not become state-of-art until the work in [6] demonstrated large improvements on the Switchboard task.

In our experiments, the MLP (also known now as DNN) is trained using standard back-propagation and the objective function is Cross-Entropy. We use the *sigmoid* activation function for the hidden layers and *softmax* for the output layer. The MLP is grown layer-wise with one pass over the training data for each layer growing phase as described in [7]. The preprocessing of the training data consists of speaker based mean and variance normalization and the training data is organized into 50 hour chunks, where each chunk is frame-level randomized.

4.2. Weight Sharing and Shift Invariance with CNNs

A regular MLP is fully connected, i.e. each hidden unit has connections to all input units. Rumelhart et.al. discussed in [8] a different type of Neural Network that uses only a subset of inputs in the form of localized receptive fields. In order to achieve shift invariance, the weight learning was changed such that the weight changes were averaged over the receptive fields.

For speech recognition, the Time Delay Neural Network (TDNN) [9] uses the concept of weight sharing and shift invariance in the temporal domain. In image recognition, Convolutional Neural Networks (CNN) [10] apply the same concepts to obtain shift

Layer	inputN x outputN
#0	243 x 128
#1	1536 x 256
#2	1280 x 2048
#3	2048 x 2048
#4	2048 x 2048
#5	2048 x 2048
#6	2048 x 7000

Table 1. CNN structure and dimensions of layer weights

Channel	A	B	C	D	E	F	G
GMM	54.7	75.3	63.8	56.8	80.6	65.2	46.6
MLP	46.2	72.4	57.4	52.1	78.4	56.7	37.8
CNN	45.8	70.7	54.8	50.9	72.1	52.5	37.9

Table 2. WER comparison of GMM, MLP, CNN

invariance in two dimensions. In [11, 12], a CNN is used as a replacement of GMMs in a HMM system. In this work, the CNN's job is to guard against changes in the spectrum, and keep the HMM to deal with temporal invariance.

Input features are 32-dimensional VTL-warped *logmel* features with a context of 11 frames. In addition to the *logmel* features, we use Δ and Δ, Δ features. The input features are mean and variance normalized at speaker level. With a window size of $9x9$ we obtain $9x3$ windows and each window contains $3x9x9$ features. The second layer of the network is also a convolutional layer, that uses a sliding window of $3x4$ over the outputs of the first layer. In total, the network has 7 layers and the first two layers are convolutional as described above. The network structure is summarized in Table 1. A more detailed model description can be found in [13].

As for regular MLPs, discriminative pre-training is applied for the CNN too. The only difference is, that the first two layers are trained together. Frame-level randomization is slightly more complicated for CNNs and requires us to write the features with sufficiently long temporal context. We also apply sequence training [14, 15] for the CNN similar to the MLP models. While the CNN is not better than the MLP for semi-clean data (channel G), we see significant improvements on all other channels. This is in line with our expectations, that the shift invariance in the feature domain helps to make the model more robust.

For the KWS task, the 1-best error rate is less important than the lattice quality. As shown in Figure 3, the neural network models not only have a better 1-best error rate, but the lattice quality is also significantly better. The baseline MLP models use different feature sets and are explained in detail in [13]. This results in better KWS performance as shown in Figure 4.

5. LATTICE GENERATION

The decoding pipeline is similar to other speaker adaptive systems with multiple decoding passes. While we do not adapt the NN models themselves, we use speaker adapted features. For the CNN, we use VTL-warped logmel features, while the regular MLPs use both VTLN and FMLLR for the input features.

The first decoding pass uses a speaker independent (SI) MLP model. The output of this decoding pass is used to estimate VTLN and FMLLR transforms and fed directly to the speaker adaptive NN

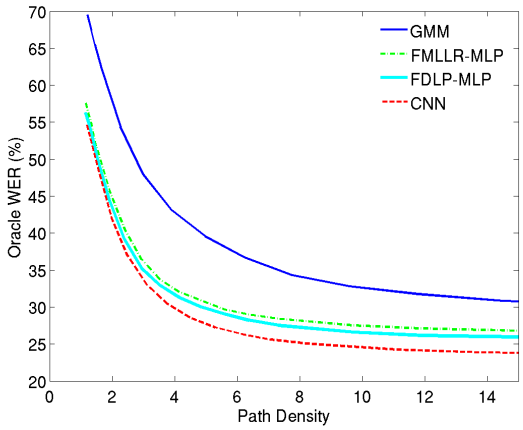

Fig. 3. Lattice Oracle Rates Comparing GMMs, MLPs and CNNs

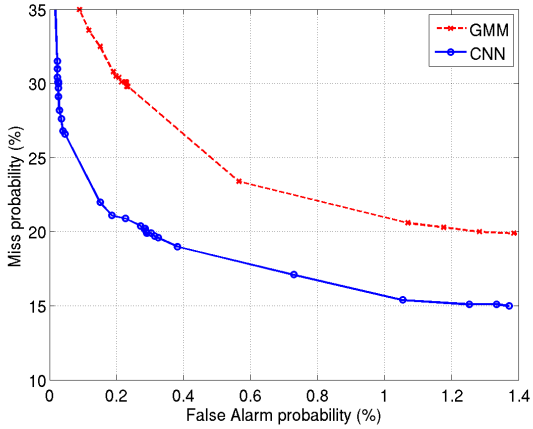

Fig. 4. Farsi KWS performance comparing GMMs and CNNs

models. This is in contrast to our Phase 1 models, where we used an SI GMM model as the first pass. Given that the error rate of the SI GMM models was very high, multiple decoding and re-adaptation passes were necessary. By replacing SI GMM first pass models with the much better SI NN models, we could estimate VTLN and FMLLR directly, therefore eliminating the need for two intermediate decoding steps.

The second decoding pass uses speaker adaptive MLPs and CNNs and produces the final lattices used for keyword search. The lattice generation uses the dynamic decoder described in [16] and converts an enriched backpointer strucure to a lattice that allows for inserting extra arcs not visited during search.

6. KEYWORD SEARCH SYSTEM OVERVIEW

In this section we describe our overall keyword search framework, which runs many KWS systems in parallel and combines the results to produce a ranked list of term occurrences. In each KWS component the audio is processed by an ASR system that produces a word lattice for each audio segment. The lattices are converted to a WFST-based index [17, 18, 19] that is used to find the location and score of a detection. The KWS components differ only in their ASR models

and audio segmentations.

6.1. Indexing

Prior to indexing and search, the word lattices are converted into phonetic WFSTs in which the input labels are phones, the output labels are the starting times of phones, and the costs are phone negative log posteriors. The resulting utterance WFSTs are used (1) as the starting point for creating the index, and (2) for retrieving the time marks of the hits during the search phase. The index WFST is the union of all the utterance WFSTs. The algorithm for converting the utterance WFSTs to a WFST index is described in [17].

6.2. Search

In WFST-based keyword search, the query WFSA is constructed using the pronunciation dictionary for IV queries and a letter-to-sound model for OOV queries. Multiple pronunciations are compactly represented in the WFSA. In both our Levantine Arabic and Farsi systems, the pronunciations are grapheme based: the pronunciation of a word is its letter sequence. Therefore, the pronunciations for both IV and OOV words can both be generated in the same way. A fuzzier search, which may improve recall while degrading precision, can be accomplished using query expansion. Specifically, we estimate the probabilities of phone-to-phone confusions and create a confusability model implemented as a phone-to-phone transducer $P2P$. Confusion probabilities are estimated using an alignment of the reference and the decoded output. Given a query q, the query WFSA Q is obtained by composing the query WFSA with the $P2P$ transducer. Varying the number of hypotheses kept after the composition ($NbestP2P$) controls the degree of query expansion, trading off between precision and recall.

6.3. Producing the final hit list

We use a 2-step search [20, 19] in which we first find the lattices containing the query through composition of the query WFST Q with the index and then use the relevant utterance WFSTs from Section 6.1 to obtain the start and end time information for the hits.

6.4. KWS system combination

After producing a list of hits and the associated scores for each KWS branch, the results are merged. For each keyword, we take the union of all hits from all systems and then produce a final list using the following procedure: (1) a hit which does not overlap with any other hit is copied to the final list, while (2) a set of overlapping hits corresponding to the same keyword is merged into one hit in the final list which has the time marks of the highest scoring hit and a score that is the sum of the hit scores. After producing the new list of hits, we normalize the scores per keyword: for each keyword we sum all the scores for all the hits and divide each score by this sum. This last normalization step produces significant improvements.

7. EXPERIMENTAL SETUP

In this section we describe the Levantine and Farsi KWS systems we deployed in the 2013 RATS evaluation.

7.1. Levantine System

We combined the 5 systems shown in Table 3. They differ in the acoustic model and audio segmentation used for producing the search lattices. In the Phase 1 evaluation system we used a phone confusion model mostly for the OOV queries, while in Phase 2 we use the same query expansion ($NbestP2P = 1000$) for both IV and OOV queries. This decision results in the same index and processing pipeline for all queries regardless of their IV/OOV status, while improving the KWS results as shown in Figure 5. As mentioned in

System	AM	Segmentation
Sys1	CNN	S1
Sys2	DNN	S1
Sys3	DNN	resegmented S2
Sys4	GMM	resegmented S2
Sys5	CNN	resegmented S2

Table 3. The 5 systems used in the Levantine Phase 2 RATS evaluation

Section 2, the evaluation metric is pFA@20%pMiss. Figure 6 shows the performance of the 5 components and the combined output on `dev-2`. It can be seen that the CNN systems are much better than the others, and that a large gain can be obtained by combining the posting lists from the individual components. Table 4 shows the KWS results for the 2 development sets `dev-1` and `dev-2` as well as the sequestered progress set.

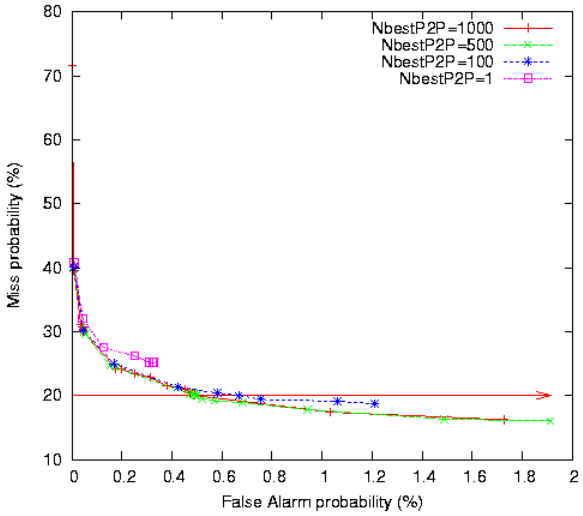

Fig. 5. Comparison of various query expansion degrees ($NbestP2P$) when applied to both IV and OOV queries

	`dev-1`	`dev-2`	`Progress`
Phase 1	0.330%	1.62%	0.29%
Phase 2	0.065%	0.295%	0.08%

Table 4. pFA@20%pMiss for Phase 1 and 2 RATS evaluation measured on development and evaluation sets.

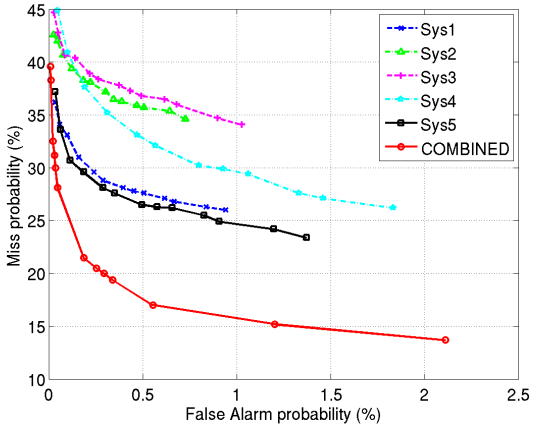

Fig. 6. Comparison of the 5 systems and the final combined output for Levantine

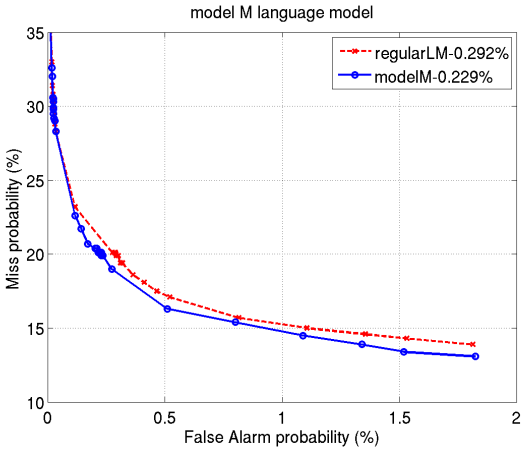

Fig. 7. Effect of Model M LM on KWS.

7.2. Farsi System

For Farsi, we generally followed the same strategies as for Levantine, combining diverse systems that differ in the audio segmentation and acoustic model. There are a few differences, which are described in this section. Instead of 5 systems, we combined 4 systems, as shown in Table 5. Because a portion of the data came with partially vowelized transcripts, we trained acoustic and language models using these (semivowelized) transcripts. Removing the diacritics not usually found in written text, we also trained unvowelized models. The GMM acoustic models were significantly worse than NN models for KWS, so we did not use a GMM system for Farsi during the Eval.

System	Vowelization	AM	Segmentation
Sys1	Unvowelized	CNN	S1
Sys2	Semivowelized	DNN	S1
Sys3	Semivowelized	CNN	resegmented S2
Sys4	Unvowelized	DNN	resegmented S2

Table 5. The 4 systems used in the Farsi RATS evaluation

For one of the systems (Sys3), we used model M, a class-based exponential language model, instead of traditional n-gram LM. Figure 7 shows the improvement in KWS performance. While there is little difference at 30%pMiss, pFA@20%pMiss is reduced from 0.292% to 0.229%, representing a 22% relative improvement.

Table 6 and Figure 8 show the performance of the individual systems as well as their combination. The WER shown is the WER over all the dev data, including channels A-H. The KWS performance is somewhat correlated with the WER, especially for pFA@30%pMiss. pFA@20%pMiss is less correlated, with Sys4 yielding unusually good KWS result. The variation in KWS results is also much larger than the differences in WER results; note that KWS results are based on a very small number (200) of selected key phrases. Combining the posting lists from the individual systems yields a large improvement in pFA@20%pMiss, from the best individual system performance of 0.228% to 0.058%, a relative reduction of 75%.

For each system, the posting list was generated using query expansion parameter $NbestP2P$=1000, like in the Levantine system. We tried another experiment where we also generated another posting list using $NbestP2P$=1 for each system. This is equivalent to a stricter match, where each hypothesized keyword hit in the posting list derives from a partial path in the decoded lattice with an exact phone sequence match to the keyword. The four $NbestP2P$=1000 posting lists and the four $NbestP2P$=1 posting lists are then combined. Figure 9 and the last lines in Table 6 show the comparison of the 4-way combination of $NbestP2P$=1000 posting lists and the new 8-way combination. pFA@20%pMiss is improved from 0.058% to 0.047%, a 19% relative reduction. Note that since audio indexing is shared, little additional computation is required, only during the keyword search step. This was the final system we used for the Phase 2 evaluation. On the sequestered progress set, this system achieved pFA@20%pMiss=0.15%.

System	WER(A-H)	pFA@30%pMiss	pFA@20%pMiss
Sys1	71.6%	0.043%	0.488%
Sys2	69.8%	0.035%	0.317%
Sys3	68.0%	0.024%	0.229%
Sys4	70.2%	0.037%	0.228%
Combo		0.012%	0.058%
+P2P1		0.011%	0.047%

Table 6. Farsi dev set performance of individual systems and combination

8. CONCLUSION

In this paper we present a state-of-the-art KWS system which had the best performance in the Phase 2 DARPA RATS evaluation. The key messages we want to highlight are:

1. CNN acoustic models perform very well for keyword search in noisy environment
2. Audio segmentation plays a significant role for keyword search
3. Large gains are obtained by combining systems which differ in the acoustic models and audio segmentations

The final combined system is 3 times better than the best component, and Phase 2 KWS system is 4-5 times better than the system fielded in the Phase 1 evaluation while being much lighter in computational resources due to the new simplified decoding pipeline.

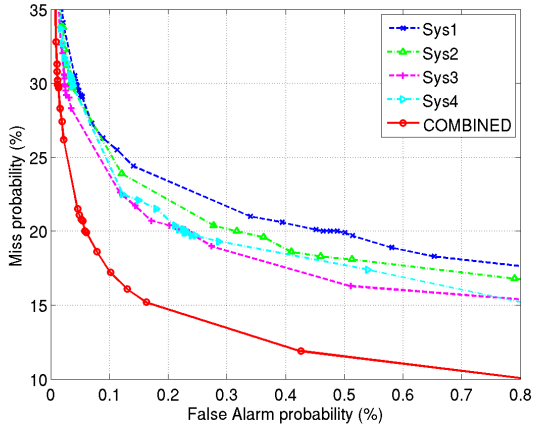

Fig. 8. 4 Farsi systems and combined output

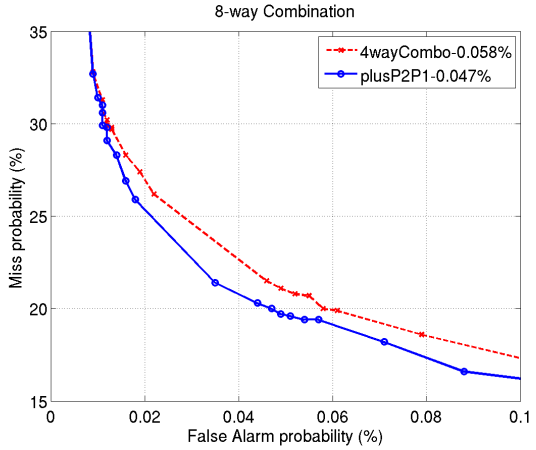

Fig. 9. 8-way combination with $NbestP2P$=1000 and $NbestP2P$=1 posting lists.

9. ACKNOWLEDGMENT

This work was supported by the DARPA RATS Program. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. Approved for Public Release, Distribution Unlimited.

10. REFERENCES

[1] J. Fiscus, J. Ajot, J. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR*, 2007, pp. 51–57.

[2] M. Maamouri *et al.*, "LDC2006S29, Arabic CTS Levantine QT training data set 5," in *Linguistic Data Consortium*, 2006.

[3] D. Graff, S. Sessa, S. Strassel, and K. Walker, "RATS data plan," in *Linguistic Data Consortium, Tech. Rep.*, 2011.

[4] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proc. ICASSP*, 2013.

[5] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection for the DARPA RATS program," in *Proc. Interspeech*, 2013.

[6] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Interspeech*, 2011.

[7] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011.

[8] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, 1986.

[9] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition: Neural networks vs hidden Markov models," in *Proc. ICASSP*, 1988.

[10] Y. L. Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. NIPS*, 1990.

[11] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition," in *Proc. ICASSP*, 2012.

[12] T. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. Interspeech*, 2013.

[13] H. Soltau, H.-K. Kuo, L. Mangu, G. Saon, and T. Beran, "Neural network acoustic models for the DARPA RATS program," in *Proc. Interspeech*, 2013.

[14] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012.

[15] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP*, 2009.

[16] H. Soltau and G. Saon, "Dynamic network decoding revisited," in *Proc. ASRU*, 2009, pp. 276–281.

[17] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata - application to spoken utterance retrieval," in *Proc. HLT/NAACL*, vol. I., 2004, pp. 33–40.

[18] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar, "Effect of pronunciation on OOV queries for spoken term detection," in *Proc. ICASSP*, 2009, pp. 3957–3960.

[19] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *Proc. ASRU*, 2009, pp. 404–409.

[20] S. Parlak and M. Saraçlar, "Spoken term detection for Turkish broadcast news," in *Proc. ICASSP*, 2008.