END-TO-END SPEECH RECOGNITION AND KEYWORD SEARCH ON LOW-RESOURCE LANGUAGES

Andrew Rosenberg, Kartik Audhkhasi, Abhinav Sethy, Bhuvana Ramabhadran, Michael Picheny

IBM TJ Watson Research Center
Yorktown Heights, NY, USA

ABSTRACT

In recent years, so-called, "end-to-end" speech recognition systems have emerged as viable alternatives to traditional ASR frameworks. Keyword search, localizing an orthographic query in a speech corpus, is typically performed by using automatic speech recognition (ASR) to generate an index. Previous work has evaluated the use of end-to-end systems for ASR on well known corpora (WSJ, Switchboard, TIMIT, etc.) in high-resource languages like English and Mandarin. In this work, we investigate the use of Connectionist Temporal Classification (CTC) networks, recurrent encoder-decoders with attention, two end-to-end ASR systems for keyword search and speech recognition on low resource languages.

We find end-to-end systems can generate high quality 1-best transcripts on low-resource languages, but, because they generate very sharp posteriors, their utility is limited for KWS. We explore a number of ways to address this limitation with modest success. Experimental results reported are based on the IARPA BABEL OP3 languages and evaluation framework. This paper represents the first results using "end-to-end" techniques for speech recognition and keyword search on low-resource languages.

Index Terms— keyword search, end-to-end speech recognition, CTC, attention networks

1. INTRODUCTION

Speech recognition is a sequence to sequence classification problem. A sequence of acoustic observations $X = \{x_1, \ldots, x_T\}$ needs to be mapped to a sequence of words $W = \{w_1, \ldots, w_N\}$, solving $P(W|X)$. Traditional speech recognition systems operate by aligning acoustic observations to some segment (phonemes, context dependent phones, graphemes, etc.). The acoustic model generates segment hypotheses for each frame. These frame level hypotheses are then mapped to word hypotheses using a hidden markov model (HMM) and viterbi algorithm incorporating a pronunciation model (lexicon) and language model.

End-to-end speech recognition approaches address the alignment problem differently, incorporating it into the optimization framework. In this work, we explore two approaches to end-to-end speech recognition: Connectionist Temporal Classification (CTC) (Section 2.1), and RNN Encoder-Decoders with Attention (Attention) (Section 2.2). Both of these address the sequence to sequence alignment problem directly in a neural network (Section 2.3). End-to-end systems are appealing because 1) they promise to simplify training, 2) they omit the iterative re-alignment process potentially leading to globally optimal solutions, 3) they allow for more sophisticated sequence models, in traditional systems sequence modeling is limited to the HMM and other related variants. End-to-end ASR has been applied to high resource languages like English and Mandarin and well known corpora, including WSJ, Switchboard TIMIT, and VoxForge ([1, 2, 3, 4, 5]). In this work, we evaluate the use of CTC and Attention to low-resource languages as part of the IARPA BABEL program (Section 3). We find that both systems are able to generate competitive word error rates (Section 4).

The IARPA BABEL program is focused the problem of Keyword Search (KWS), the task of localizing an orthographic (written) query in a collection of speech. The simplest approach to KWS is to use speech recognition to generate a (1-best) transcript of the speech corpus, and treat this as a text search problem (ctl-f, grep, etc.) [6, 7]. However, even state-of-the-art speech recognition systems generate a significant amount of errors ($\sim$6-7% WER [8, 9]). Therefore all state-of-the-art KWS approaches maintains a collection of less confident hypotheses rather than generating a 1-best transcription, using ASR to generate a lattice or confusion network of word hypotheses and using this lattice (or some derivative representation) as a search index. Moreover, a variety of approaches to address is problem have been proposed including query expansion [7], and subword indexing where the index comprises, phone, morph or syllable units [10, 11, 12]. For ASR to be an effective tool for KWS it needs to generate high quality transcription, but it also needs to maintain a rich hypothesis space. In this work, we evaluate the efficacy of end-to-end ASR systems to the KWS index generation problem. Despite their competitive ASR performance, we find both end-to-end systems to generate very peaky posteriors, resulting in poor KWS performance (Section 5). We then investigate the use of these systems for representation learning, using embeddings from their operation as input to a more traditional HMM-DNN ASR/KWS system, finding them to generate improved performance but still failing to surpass a competitive traditional ASR system (Section 6).

2. END-TO-END SPEECH RECOGNITION

In this section, we define Connectionist Temporal Classification (CTC) Networks and the RNN Encoder-Decoder with Attention (Attention-ASR). Both approaches address a sequence classification problem mapping a sequence of acoustic observations $X = \{x_1, \ldots, x_T\}$ to a sequence of words $W = \{w_1, \ldots, w_N\}$. Both approaches will address this by solving a slightly simpler problem, mapping the sequence of acoustic observations to a sequence of graphemes $Y = \{y_1, \ldots, y_L\}$. In both cases, the mapping from graphemes to words is solved by an inverted dictionary, and language model. Though not the case for all sequence classification problems, for speech recognition, we assume that $N > L > T$. Critical to this problem is solving an alignment problem, mapping graphemes $\{y_l\}$ to acoustic observations $\{\mathbf{x}_t\}$.

2.1. Connectionist Temporal Classification Networks

The CTC Network is a recurrent neural network (RNN) [4], where it differs from a typical RNN is in its outputs and the objective function used during training. In a typical RNN, one label y is generated per frame of input x. The CTC approach to solving the sequence to sequence classification where the input sequence length T is not equal to the output sequence length L is as follows. Multiple T lengthed sequences of outputs are mapped to identical L lengthed sequences. This mapping is performed by removing all repeated labels. In this way, sequences `AAAB` and `AABB` are both mapped to the sequence `AB`. However, this would make it impossible to generate sequences that do, in fact, contain repeated labels. To address this issue, an an additional "blank" output label is included the vocabulary V. The CTC mapping function, $\mathcal{B}$, both eliminates blanks, and all repeated labels. The CTC objective function introduces a path variable π

$$p(\pi|X) = \prod_{t=1}^{T} h_{\pi_t}^t, \forall \pi \in V^T$$

, where h_k^t is the probability of observing the label k at time t, as generated from a softmax output layer. To generate a likelihood of a given label sequence l, all π's that map to the same label sequence Y are marginalized out,

$$p(l|X) = \sum_{\pi \in \mathcal{B}^{-1}(l)} p(\pi|X).$$

This marginalization is calculated via dynamic programming similar to the HMM forward-backward algorithm over a trellis of outputs. Moreover, the gradient of this likelihood function can be efficiently calculated and used to train the RNN via back-propagation through time. In this work, we use the Eesen toolkit to [5] to train CTC networks and generate posteriors, and IBM Attila Speech Recognition Toolkit [8, 13] to decode. The RNN has 4 bidirectional layers, with 200x2 LSTM units and is trained with sgd with momentum.

2.2. RNN Encoder-Decoder with Attention

RNNs have been used to solve sequence to sequence classification problems including machine translation [14] and syntactic parsing [15]. The typical approach used is to use one network to *encode* the input sequence into a fixed dimensional representation e, and a second recurrent network to *decode* e into an output sequence Y. If either sequence contains a complicated structure, it can be challenging for the network to encode all of this information into a single, fixed sized, vector. The idea of "attention" is to allow the decoder to have access to the full sequence generated by the encoder, and a third *attention* decision process is used to weight the encoded observation for at each decoder step. The attention process is itself a recurrent network. In this way, the attention decoder simultaneously is determining what part of the input is most important to generate an output label, and what that label should be. The attention encoder-decoder has been used for speech recognition in previous work [2, 3]. Our approach follows an implementation[1] described in [16, 2].

Specifically,

$$y_i = Decoder(s_{i-1}, g_i)$$

$$s_i = Recurrency(s_{i-1}, g_i, y_i)$$

$$g_i = \sum_{j=1}^{L} \alpha_{ij} e_j$$

$$\alpha_{ij} = exp(h_{ij}) / \sum_{j=1}^{L} exp(h_{ij})$$

$$h_{ij} = Score(s_{i-1}, e_j)$$

where e_j are the outputs of the encoder bidirectional RNN. From bottom to top, $Score(s_{i-1}, e_j)$ is a MLP function assigning the raw attention scores h_{ij}. A softmax function converts these to attention weights, α_i. These are used to weight the encoded features resulting in g_i, which are then used as input to $Recurrency$ a unidirectional RNN responsible for generating the output sequence elements y_i via $Decoder$, a readout function of the previous recurrent state s_{i-1} and current attended encoding g_i.

The encoder is a 4 layer Bidirectional GRU RNN of 320x2 hidden units. The two deepest layers concatenate adjacent frames. The decoder is a unidirectional GRU RNN also with 320 hidden units. Training uses *adadelta* and momentum for training with gradient clipping at 10.0. The training schedule comprises 1 pretraining iteration, 10 main iterations, and two annealing phases of 3 iterations each with reduced initial learning rates.

2.3. Comparing end-to-end approaches

Critical to any sequence to sequence classification problem is how to handle the alignment from input to output sequence. Traditional speech recognition takes an iterative approach to this problem. In a "flat-start" system, initial (naive) alignment hypotheses are used to train a seed acoustic model which performs frame-level classification of each acoustic frame to an output label, this model is then use to re-align the training data, and another acoustic model is trained with this newly aligned data. This process is repeated, in many cases, using more and more sophisticated acoustic models. This can be viewed (without statistical formalism) like a majorization-minimization algorithm where a latent alignment variable λ is introduced. Frame-level performance (accuracy or model likelihood) is optimized given a fixed λ, then the improved frame-level model is used to update λ.

Both of end-to-end approaches can also be interpreted as introducing an alignment variable and solving the alignment problem explicitly during training.

CTC introduces an alignment variable, by way of the path variable π, and then marginalizes across all paths.. Dynamic programming provides an efficient approach to calculating an exponential number of alignments and marginalizing over their probability.

Training attention and decoder components allows the RNN objective function to solve the alignment and labeling problems simultaneously. Each output y_i is a the results of a function of the input sequence X, and the previous output y_{i-1} using an intermediate function that (in effect) localizes the important subsequence $X' \subset X$ and weights its contribution. The alignment variable in this case is the attention weights mapping each output symbol y_i to X'_i.

There are differences in the acoustic models used each system mapping acoustic observations to an output labels (grapheme, CD state, etc.). In particular, the CTC addition of the blank symbol leads to very different hypotheses. However, a major difference between these is in how they address the alignment problem.

3. MATERIAL AND EVALUATION

This work was performed as part of the IARPA BABEL Program. This program is focused on keyword search and speech recognition on low-resource languages. The investigated languages and language pack identifiers are described in Table 1. For each language, training and development set partitions are defined by the language

[1] `https://github.com/rizar/attention-lvcsr`

pack distribution. Training sets are approximately 40 hours, with 15 hour development sets. For all experiments the input acoustic features are multi-lingual features trained on 24 BABEL languages (ML24). These features are trained via multi-task training in a two stage DNN as described in [17].

Language	ID	Language Pack ID
Pashto	104	IARPA-babel104b-v0.4bY
Guarani	305	IARPA-babel305b-v1.0c
Igbo	306	IARPA-babel306b-v2.0c
Amharic	307	IARPA-babel307b-v1.0b
Mongolian	401	IARPA-babel401b-v2.0b
Javanese	402	IARPA-babel402b-v1.0b
Dholuo	403	IARPA-babel403b-v1.0b
Georgian	404	IARPA-babel404b-v1.0a

Table 1. *Investigated Languages and Identifiers*

Within this program and corresponding evaluation, keyword search is a query detection task, with performance evaluated using Term-Weighted Value (TWV) [18]. TWV is defined as:

$$TWV(\theta) = 1 - [P_{Miss}(\theta) + \beta \cdot P_{FA}(\theta)]$$

where $\beta = 999.9$, P_{Miss} is the miss rate averaged across all keywords, and P_{FA} is the average false alarm rate. The denominator of P_{FA} is defined for each keyword as $T - N_{true}(kw)$, where T the amount of evaluation speech and N_{true} is the number of true hits.

Maximum Term Weighted Value (MTWV) is defined as the TWV obtained from optimal setting of a decision threshold, θ. In this work, we report MTWV so as to disentangle hypothesis ranking from the threshold identification problem.

4. ASR PERFORMANCE

We first report the 1-best WER performance of each system on the corresponding development set for each language in Table 2. For all languages other than Georgian (404) evaluation was performed on manually segmented utterances with silent regions ignored. For Georgian, automatically segmented utterances were used. Moreover, when decoding 404, additional material collected from the web [19] was used for language modeling. We compare these with a HMM-DNN [8] trained with hessian free sequence training [20].

Language	ID	HMM-DNN	CTC	Attn
Pashto	104	52.7	52.8	55.5
Guarani	305	50.5	51.7	53.8
Igbo	306	61.4	64.2	62.1
Amharic	307	46.5	51.7	52.6
Mongolian	401	56.8	59.8	62.6
Javanese	402	57.3	58.3	64.6
Dholuo	403	41.6	44.2	48.2
Georgian	404	41.8	49.1	57.8

Table 2. *1-best WER using CTC, Attention, and HMM-DNN*

In general, we find that CTC performs slightly worse than the HMM-DNN trained with cross-entropy, and slightly worse than one trained with sequence training. The Attention results are typically worse than CTC or either hybrid system, though these systems were more coarsely tuned for optimal language model weights at decoding. This was an impact of Attention decoding being more computationally intensive. Even with this caveat, the discrepancy between the HMM-DNN system and the Attention approach on Georgian (404) is surprisingly high. Georgian is a highly agglutinative language – one hypothesis is that this created more problems for the Attention decoder than other languages did.

5. USING END-TO-END ASR FOR KWS

Following an approach described in [21], we perform keyword search on lattices generated by the two end-to-end systems. The general KWS approach is to generate lattices for all of the data to be searched over. These lattices are used as the search index for all queries made up of seen, in-vocabulary, terms. All queries containing out-of-vocabulary terms are searched for in a graphemic index. Scores for each query are normalized via, sum-to-one normalization.

On Pashto (104), the CTC-generated lattices result in an MTWV of 0.29, compared to a baseline of 0.38 from the HMM-DNN system trained on the same input features. This is dramatically worse considering that the two systems have almost identical WER performance (CTC:52.8; DNN-HF: 52.7). Though not completely predictive, ASR performance has been a reasonable, if rough, proxy for KWS performance. We are left with the question of why, given the reasonably good (considering the amount of data and challenges of the language), and comparable ASR performance, KWS performance is so poor on lattices generated by end-to-end ASR.

High quality KWS, especially with highly errorful transcription, requires a rich hypothesis space to identify lower confidence hits. With WER scores between 40 and 60, a high-quality transcript is not sufficient for effective KWS, many hits receive low confidence, but will remain above a KWS decision threshold. To help explain this performance discrepancy we investigate the distribution of graphemic posteriors of the three systems on Igbo (306). The entropy of posteriors is used as a representation of how varied hypothesis space each system is capable of delivering. The CTC and Attention numbers are calculated over the graphemic vocabulary (omitting the blank for CTC, while including the *space* character for Attention) while the HMM-DNN calculation is over 6000 Context Dependent phones. The CTC average entropy is 0.331, while Attention is 0.198. This is in comparison to an HMM-DNN average entropy of 0.678. While the one best ASR performance of the end-to-end systems is competitive with the HMM-DNN, the hypothesis space is very sparse with "peaky" (i.e. low entropy) posteriors resulting in thin lattices.

6. END-TO-END SYSTEMS AS FEATURE EXTRACTORS

ASR results (cf. Table 2) suggest that the end-to-end systems are able to learn effective mappings from input features to graphemes. In this section, we investigate whether these mappings more informative than the original features. To do this, we investigate the use of end-to-end systems as feature extractors.

For the CTC system, we train a new version with a final hidden layer that has 50 bidirectional units (100 total), and extract activations from this hidden, encoder layer. We use these encoded features to train an HMM-DNN system, Hybrid-CTC, on the development languages (all but 404-Georgian). These results would be able to tell us if the CTC encodings, based on ML24 features, are any more effective than the originals (with results reported in Table 2). This system is trained with the IBM Attila Speech Recognition Toolkit. Results can be found in Table 3.

The Hybrid-CTC systems yield much more competitive KWS results than the pure CTC variant on 104 (0.34 vs. 0.29). Also, the ASR performance is improved across all languages. However, despite improved WER, the MTWV of the Hybrid-CTC is worse

ID	HMM-DNN		Hyb-CTC	
	WER	MTWV	WER	MTWV
104	52.7	0.3853	51.0	0.3447
305	50.5	0.5345	47.7	0.5171
306	61.4	0.3211	60.3	0.3103
307	46.5	0.5952	45.7	0.5604
401	56.8	0.4674	54.6	0.4212
402	57.3	0.4503	55.1	0.4209
403	41.6	0.6035	41.0	0.5705

Table 3. *WER and MTWV using HMM-DNNs trained on ML24, and CTC-encoded features*

than that obtained from the HMM-DNN trained on original ML24 features. This suggests that the "peakiness" may not only impact posteriors, but aspects of the encoded features as well.

We then inspect the use of Attention models as a feature extractor. We perform this investigation on 306-Igbo, since it was the most difficult language (highest WER and lowest MTWV). For the feature transformations, we use the activations from the bidirectional encoder. These have 320 bidirectional units (640 total). However, the Attention encoder subsamples the input frames twice resulting in one encoded frame per four input frames. This is done to reduce the computational complexity of the encoder, but it also tends to generate improved performance. To reconstruct the original frame size, adjacent frames are linearly interpolated. As before we use IBM Attila to train a Hybrid-Attention system on 306. This results in a WER of 59.6 and MTWV of 0.2967. This is somewhat consistent with the Hybrid-CTC result, we see improvements to WER over both the baseline features and the pure Attention system, but rather poor MTWV even using the encoded features.

Inspecting the posterior entropy on Igbo (306) as in Section 5, we find the Hybrid-CTC system has an average entropy of 0.581, and the Hybrid-Attention system yields 0.340 in comparison to the HMM-DNN value of 0.678. Here all numbers are computed over the same 6,000 Context Dependent states. This suggests that the hybrid model substantially smooths the model posteriors, modestly *improving* ASR performance, and resulting in improved KWS performance compared to the original CTC system. While using the end-to-end systems as feature encoders helps WER performance, MTWV results are still better with the original features. This is consistent with the fact that posteriors generated by these systems are not as diverse (high entropy) as those generated from the original HMM-DNN.

We then combine these encoded features with multilingual acoustic features generated by RWTH Aachen [22] as input to a DNN trained with hessian-free sequence training. CTC+RWTH systems are trained only for the most difficult languages, while Attention+RWTH systems were trained for all languages. This is a system combination approach exploited by a number of sites during the IARPA BABEL evaluation to leverage complementary feature representations (e.g. [23]). In Table 4, we report ASR and KWS performance of the resultant systems. Across all systems, we find that the initial ML24 features to combine most effectively with the RWTH features. While we find that the end-to-end transformations result in improved WER in isolation (Table 3), these transformed features do not combine better with additional features via early fusion, nor do they improve MTWV performance.

7. CONCLUSION

End-to-end speech recognition systems have emerged as competitive alternatives to traditional ASR approaches. They have significant advantages in terms of simplicity of training and the ability to incorporate more sophisticated sequence models. In this work we evaluate the utility of CTC and RNN Encoder-Decoders with Attention, two end-to-end systems, for recognition of speech from low-resource languages with limited available training data. This paper represents the first results using "end-to-end" techniques for speech recognition and keyword search on low-resource languages. We find that they are able to generate competitive results, though fail to surpass the WER obtained by a more traditional HMM-DNN approach.

ID	ML24+RWTH		CTC+RWTH		Attn+RWTH	
	WER	MTWV	WER	MTWV	WER	MTWV
104	47.9	0.4088	49.3	0.3775	50.5	0.3528
305	46.6	0.5437	–	–	46.8	0.5201
306	59.1	0.3369	59.1	0.3211	59.3	0.2984
307	42.2	0.6060	–	–	44.3	0.5441
401	52.4	0.4860	52.1	0.4724	53.7	0.4436
402	53.7	0.4708	53.6	0.4554	54.4	0.3974
403	39.6	0.6030	–	–	40.3	0.5761
404	40.8	0.6979	40.6	0.6889	43.9	0.6541

Table 4. *WER and MTWV using CTC, Attention, and baseline ML24 features in combination with RWTH features*

We also investigate the use of these approaches for keyword search. We find their use to be limited in this regard. Both systems generate posteriors that are very low-entropy, resulting in a limited hypothesis space for KWS. We explored two approaches to improve their performance both using the end-to-end systems as feature encoders, building HMM-DNN systems using their encoders as input. This results in improved performance to KWS in both cases, but again fails to surpass the HMM-DNN baseline. Finally we investigate their use in system combination, to determine if the feature encoding is more or less complementary with an externally constructed feature representation. We find that in combination, end-to-end transformed features to perform worse than the original features.

These results point to a few directions to improve end-to-end systems for ASR and KWS. First, the encoder in both CTC and Attention models can be enhanced. The possibilities here are vast; most improvements to DNN acoustic modeling can be applied to end-to-end encoders (e.g. CTC criteria applied to CNN acoustic models [24]). The end-to-end decoders may need more specific modification to generate a more diverse hypothesis space for effective KWS. More aggressive regularization may be useful here, as would direct incorporation of a language model and other side information to promote maintenance of alternate hypotheses.

ACKNOWLEDGMENTS

We thank our partners at RWTH Aachen for help in providing additional acoustic features. This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. 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.

8. REFERENCES

[1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015.

[2] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," *CoRR*, vol. abs/1508.04395, 2015.

[3] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, New York, NY, USA, 2006, pp. 369–376, ACM.

[5] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *ASRU*, 2015.

[6] Jonathan Mamou, Bhuvana Ramabhadran, and Benjamin Sznajder, "Combination of multiple speech transcription methods for vocabulary independent search," in *in Searching Spontaneous Conversational Speech Workshop, SIGIR*, 2008, pp. 20–27.

[7] Murat Saraclar, Abhinav Sethy, Bhuvana Ramabhadran, Lidia Mangu, Jia Cui, Xiaodong Cui, Brian Kingsbury, and Jonathan Mamou, "An empirical study of confusion modeling in keyword search for low resource languages," in *2013 IEEE ASRU*, Olomouc, Czech Republic, December 2013, pp. 464–469.

[8] George Saon, Tom Sercu, Steven J. Rennie, and Hong-Kwang Jeff Kuo, "The IBM 2016 english conversational telephone speech recognition system," *CoRR*, vol. abs/1604.08242, 2016.

[9] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "The Microsoft 2016 Conversational Speech Recognition System," *ArXiv e-prints*, Sept. 2016.

[10] Hang Su, James Hieronymus, Yanzhang He, Eric Fosler-Lussier, and Steven Wegmann, "Syllable based keyword search: Transducing syllable lattices to word lattices," in *IEEE SLT 2014*, South Lake Tahoe, NV, December 2014, pp. 489–494.

[11] Karthik Narasimhan, Damianos Karakos, Richard M. Schwartz, Stavros Tsakalidis, and Regina Barzilay, "Morphological segmentation for keyword spotting," in *EMNLP 2014*, Doha, Qatar, October 2014, pp. 880–885.

[12] William Hartmann, Viet Bac Le, Abdelkhalek Messaoudi, Lori Lamel, and Jean-Luc Gauvain, "Comparing decoding strategies for subword-based keyword spotting in low-resourced languages.," in *INTERSPEECH*, 2014, pp. 2764–2768.

[13] Hagen Soltau, George Saon, and Brian Kingsbury, "The IBM attila speech recognition toolkit," in *IEEE SLT 2010*, Dilek Hakkani-Tür and Mari Ostendorf, Eds., Berkeley, California, December 2010, pp. 97–102, IEEE.

[14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[15] Richard Socher, Christopher Manning, and Andrew Ng, "Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks," in *NIPS*2010 Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[16] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *CoRR*, vol. abs/1506.07503, 2015.

[17] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nußbaum-Thom, Michael Picheny, Zoltán Tüske, Pavel Golik, Ralf Schlüter, Hermann Ney, Mark J. F. Gales, Kate M. Knill, Anton Ragni, Haipeng Wang, and Philip C. Woodland, "Multilingual representations for low resource speech recognition and keyword search," in *IEEE ASRU 2015*, Scottsdale, AZ, December 2015, pp. 259–266, IEEE.

[18] Jon Fiscus, "The 2006 spoken term detection evaluation plan," http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf.

[19] Gideon Mendels, Erica Cooper, Victor Soto, Julia Hirschberg, Mark Gales, Kate Knill, Anton Ragni, and Haipeng Wang, "Improving Speech Recognition and Keyword Search for Low Resource Languages Using Web Data," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015.

[20] Brian Kingsbury, Tara N. Sainath, and Hagen Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *INTERSPEECH 2012*, Portland, Oregon, September 2012, pp. 10–13.

[21] Brian Kingsbury, Jia Cui, Xiadong Cui, M.J.F. Gales, Kate Knill, Jonathan Mamou, Lidia Mangu, David Nolden, Michael Picheny, Bhuvana Ramabhadran, Ralf Schl
"uter, A. Sethy, and P.C. Woodland, "A high-performance cantonese keyword search system," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 8277–8281.

[22] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney, "Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages.," in *INTERSPEECH*. Citeseer, 2014.

[23] William Hartmann, Le Zhang, Kerri Barnes, Roger Hsiao, Stavros Tsakalidis, and Richard Schwartz, "Comparison of multiple system combination techniques for keyword spotting," in *Interspeech 2016*, 2016, pp. 1913–1917.

[24] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, César Laurent, Yoshua Bengio, and Aaron Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Interspeech 2016*, 2016, pp. 410–414.