ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097249

BUILDING KEYWORD SEARCH SYSTEM FROM END-TO-END ASR SYSTEMS

Ruizhe Huang[1], *Matthew Wiesner*[2], *Leibny Paola Garcia-Perera*[1,2], *Dan Povey*[3],
Jan Trmal[1,2], Sanjeev Khudanpur[1,2]

[1]Center for Language and Speech Processing, Johns Hopkins University, USA
[2]Human Language Technology Center of Excellence, Johns Hopkins University, USA
[3]Xiaomi Corporation, Beijing, China

ABSTRACT

Keyword search (KWS) systems are commonly built on top of existing automatic speech recognition (ASR) systems. However, end-to-end (E2E) ASR models are not naturally equipped with word-level timing information or confidence. Existing methods for re-purposing E2E ASR systems for KWS are largely heuristic or model-*specific*. In this paper, we describe a general KWS pipeline, applicable to any ASR model that generates N-best lists. We extract timing information using either external word-aligners, or time-preserving weighted finite-state transducer-based decoders. We show that our light-weight, ASR-*agnostic* approach for confidence estimation based on N-best lists outperforms other commonly used heuristics, such as using the decoder's softmax probability, and even a more complicated dedicated confidence estimation model (CEM). Finally, we compare our performance to hybrid ASR models, extensively evaluating the impact of word-level *timing*, *confidence*, and *recall* on KWS performance. Our KWS pipeline is available online[1], suitable for evaluating the aforementioned ASR components as downstream tasks.

***Index Terms*—** speech recognition, end-to-end, keyword search, information retrieval, confidence, forced alignment

1. INTRODUCTION

Keyword search (KWS), also called spoken term detection (STD) [1], enables search of large spoken corpora such as lectures, meeting recordings, call center conversations or videos on the web. A KWS system proposes candidate matches[2] for user-specified search terms in an audio corpus.

Typical KWS systems are built on top of automatic speech recognition (ASR) systems [3, 4, 5, 6, 7, 8], which decode the input speech into possible word-sequence hypotheses before searching. There is also work that explores the possibility of ASR-free KWS [9, 10, 11]. However, in this work, we focus on ASR-based KWS, as ASR-based KWS generally outperforms ASR-free KWS. Specifically, as recent progress in end-to-end (E2E) ASR [12] has generally outpaced research into their use in KWS systems, we focus on building KWS system from E2E ASR. KWS can also help diagnosing ASR performance beyond the commonly used word error rate (WER).

To enable good KWS systems, accurate estimation of timing and confidence scores for words is required. A high recall of words in the ASR N-best or lattice outputs is also important. However, unlike hybrid ASR, which is decoded using methods based on the weighted finite state transducer (WFST) framework and provides a rich lattice where timing information and confidence scores can be easily obtained, E2E ASR results do not naturally come with timing or confidence per word. This work will address and evaluate the impact of these challenges on KWS performance.

There has been previous research on KWS based on E2E ASR. People have used either lattices [13, 4] or N-best lists [5, 6, 8] as the underlying ASR outputs. Various methods have been proposed to improve the accuracy of timing and confidence. For example, [5, 6, 8] fine-tune the ASR systems in terms of time alignment or confidence scores specifically for KWS. Our work conducts KWS on N-best lists output by E2E ASR, similar to [6, 8]. We explore additional options to get timing and confidence. More importantly, we aim to provide a general, out-of-the-box KWS pipeline without any fine-tuning, which is applicable to any ASR model capable of generating N-best lists. The timing information (e.g., [14, 15]) and confidence scores (e.g., [16, 17]) extracted by any external model can be easily plugged-in and evaluated. This facilitates evaluation of competing ASR systems with similar WERs in controlled experiments. On this test-bed, we conduct extensive intrinsic and extrinsic evaluations of the quality of the timing and confidence information obtained by different methods, yet it turns out that the bottleneck for KWS based on E2E ASR is its lower word recall.

2. KWS SYSTEM OVERVIEW

Our system takes as inputs the N-best lists, timing information and confidence scores (detailed in subsequent sections) from ASR. It then builds efficient indices on top of the inputs. During search time, for each query term, we return a putative hitlist of (utterance_id, start_time, end_time, confidence_score).

More specifically, we first greedily align each ASR hypothesis, by edit distance, to make a compact WFST representation – confusion network [18], also called "sausage". On top of this, unlike previous work [5, 6, 8], we use timed factor transducer (TFT) [19] as inverted index, following its successful applica-

[1] `https://github.com/huangruizhe/kws`

tion to conventional KWS for hybrid ASR [3]. TFT naturally supports any query expressed as WFST, *e.g.*, phrasal and wildcard KWS. Caution has to be taken when constructing TFT over confusion networks, as TFT are originally proposed for ϵ-free lattices, whereas in our case, the confusion network can contain ϵ arcs due to aligning hypothesis of different lengths. As a workaround, we explicitly treat ϵ as a non-ϵ normal symbol and translate between equivalent sequences containing ϵ.

3. EXTRACTING TIME ALIGNMENTS

We compare three ways to extract time alignments for the 1-best hypothesis. Based on the 1-best hypothesis's alignment, the sausage timing is distributed proportionately.

- **HMM-GMM aligner trained on ASR training data**: If the ASR training data is available, we can train an HMM-GMM model with, e.g., Kaldi toolkit [20], to perform forced alignment of the hypotheses. This method is considered the most accurate, and will be used as the baseline.
- **Universal external aligner**: If the language of interest is supported by an off-the-shelf universal external aligner, e.g., Montreal Forced Aligner (MFA) [21], we can use it to align the E2E ASR results with time. Note, MFA is also a HMM-GMM system, but trained on some external data different from the ASR training data. Thus, MFA can potentially suffer from data domain mismatch.
- **WFST-based decoding for CTC posteriors**. Many E2E ASR systems are based on either connectionist temporal classification (CTC) [22], RNN transducer (RNN-T) [23] or joint CTC/Attention architecture [24]. These systems can adopt WFST-based decoding [25, 26] as in hybrid ASR. More specifically, E2E ASR systems first generate frame-by-frame posteriors of the whole utterance. Then, the posteriors are composed with a WFST decoding graph to search for the best token sequence. The decoding graph ("TLG graph") incorporates the topology (T) mapping CTC alignments to tokens, lexicons (L) and language models (G). The composed graph is then determinized and minimized to improve efficiency. During decoding, the decoder will try to find the best path for the utterance through the decoding graph. The position of the input or output labels on the path can be seen as their timing.

4. ESTIMATING SAUSAGE ARC CONFIDENCE

We need a score for each arc in the sausage. More specifically, for each arc with label w in the i'th bin of the confusion network for utterance $\mathbf{x}$, we want a confidence score $p(w|\mathbf{x};i)$ to reflect how likely the word (or the arc) is correct. The question is how to compute $p(w|\mathbf{x};i)$.

- **Aggregation of sequence-level posterior probability** [27]. In E2E ASR, each hypothesized word sequence usually comes with a score, interpreted as its total log posterior probability. We can normalize the scores to sum to one among the N-best, which can be regarded as the proper posterior probability for each hypothesis. Specifically, for each hypothesis h_j with score s_j in the N-best list, let $p(h_j|\mathbf{x}) = \text{softmax}(s_j/\tau)$, where τ is a temperature scaling factor. When $\tau \to \infty$, the distribution becomes uniform; when $\tau \to 0$, the 1-best hypothesis gets all probability mass.
- **Average of the decoder's softmax probability.** Alternatively, we follow the approach in [28, 6], which is to obtain word-level posteriors by aggregating token-level posteriors output by the softmax layer in the decoder. We used max as the aggregation function. Then, the scores for the sausage arcs gone through by multiple hypotheses are obtained by further averaging the word posteriors from different hypotheses. Note that this method does not guarantee that the probability of each sausage bin sums to one.
- **Auxiliary confidence estimation model**. Following [16], we train a dedicated confidence estimation model on top of the ASR model. In fact, the model in [16] does not need to access the ASR model. It only takes the N-best lists as well as several handy scores as input. More specifically, we first merge the N-best list into a sausage. Then, each sausage arc has the following features: pre-trained word embeddings (e.g., derived from `fastText` [29]), duration, the confidence scores taken from other methods such as the above two. Then the sausage is fed into a bidirectional lattice RNN network, which outputs a score between 0 and 1 for each arc.

5. EXPERIMENTS AND ANALYSIS

We experiment on English conversational telephone speech. For ASR training, Switchboard corpus and additional Fisher text of 23M words are used. These two data sets consist of conversations between strangers about assigned topics. For KWS development and evaluation, we use two datasets. One is STD2006 Dev/Eval, for benchmarking with other KWS systems in 2006 NIST STD evaluation [30]. The other is CALLHOME English, containing spontaneous conversations between friends and families, a mismatched scenario cf. Switchboard or Fisher. Dataset info can be found in Table 1.

Dataset	#Hours	WER %		
		Hybrid	E2E	k2
Switchboard (Train/Eval)	260h / 4h	- / 11.4	- / 11.0	- / 11.0
STD2006 (Dev/Eval)	3h / 3h	11.4 / 13.6	10.8 / 12.0	11.0 / 12.4
CALLHOME (Dev/Eval)	3h / 1.5h	20.0 / 18.7	20.2 / 18.8	20.3 / 18.8

Table 1. Dataset statistics and word error rates of ASR models

To reflect practical deployment scenarios, we use off-the-shelf ASR models without additional KWS application-specific fine-tuning. ASR information is in Table 1. We use hybrid HMM-DNN models from Kaldi[2] [20] and E2E models from ESPnet[3] [31]. Decoders for all systems are configured to achieve similar oracle WERs on 50-best lists (Section 5.4).

[2] https://github.com/kaldi-asr/kaldi/tree/master/egs/swbd

[3] https://github.com/espnet/espnet/tree/master/egs2/swbd/asr1

KWS system	STD2006 Eval	CALLHOME Eval
Hybrid, Lattice	0.8155 / 0.9345	0.8586 / 0.9382
Hybrid, 50**-best**	0.8389 / 0.9247	0.8534 / 0.9136
$*$ **E2E,** 50**-best**	<u>0.8121</u> / 0.8972	<u>0.8031</u> / 0.8720
Fusion, 50**-best**	**0.8426 / 0.9382**	**0.8693 / 0.9449**

Table 2. KWS performance (ATWV/STWV) on Eval sets for different systems ($\Delta_T = 0.5$s). The closer to 1.0, the better. The best results are in bold font. The default setup for E2E ASR-based KWS is marked with an asterisk ($*$), and the underlined numbers will appear in other tables for reference.

Search terms can be words or phrases. STD2006 data comes with lists of search terms for both Dev and Eval sets. For CALLHOME, we choose approx. 2K words/phrases from the reference transcription with high TF-IDF scores or pointwise mutual information (PMI). Heuristic filters are applied to remove phrases that do not look like proper collocations.

The overall KWS performance will be measured by actual TWV (ATWV) and Supremum TWV (STWV). Detailed definitions can be found in NIST KWS evaluation plan[4]. When computing TWV, there is a temporal tolerance collar (Δ_T) which controls the allowable distance (in seconds) between the centers of sys/ref occurrences. The smaller Δ_T is, the stricter time alignment is required. By default, we take $\Delta_T = 0.5$ sec.

Three factors can impact KWS performance (i.e., TWV): timing, confidence scores and word recall. We will provide intrinsic and extrinsic evaluation for each of the factors in the subsequent sections, after an overall KWS evaluation.

5.1. Evaluation of KWS

First, we benchmark KWS performance of hybrid and E2E systems in Table 2. For hybrid systems, KWS can be based on N-best lists or lattices. For E2E system, by default, we take $N = 50$, use HMM-GMM aligner as in section 3, and use aggregated sequence-level posterior probability (denoted as POST) as confidence score (Section 4), where the best temperature scaling factor τ ($0.2 \sim 0.5$) is tuned from Dev data.

Note that ATWV reflects the system's actual performance in operation, while STWV is the maximum achievable ATWV when all false alarms are not penalized. Overall, although E2E ASR has better WER (Table 1), KWS based on hybrid ASR (the first two rows) outperforms its E2E counterpart (3rd row) in terms of both ATWV and STWV.

Examining the 2nd/3rd rows of Table 1, we see that the ATWV of E2E system falls behind hybrid system by a considerable margin. However, the STWV lags by *almost the same margin*. Thus, we conjecture that the worse E2E KWS performance is mainly due to its lower recall (i.e., low STWV), rather than poor score calibration. For system fusion, we combine the hitlists of the systems on row 2 and 3. It turns out to be better than both single systems. This result agrees with [5].

5.2. Evaluation of Time Alignments

The ESPnet model does not have time stamps for its ASR outputs. Thus, following Section 3, we compare several ways to obtain timing alignment. We use the `k2`[5] framework and the decoders in `icefall`[6] for WFST-based decoding.

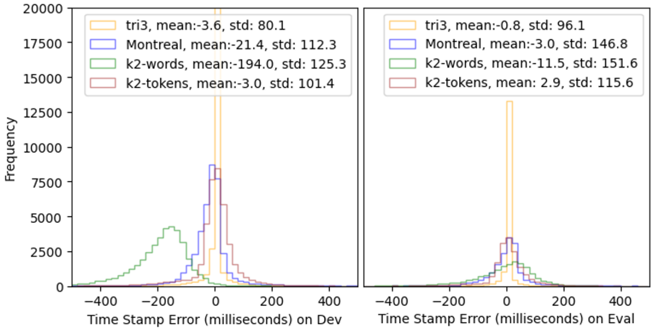

Fig. 1. Intrinsic evaluation of time alignments: comparing time stamp errors (TSE, in milliseconds) of different ways to obtain alignments for CALLHOME E2E ASR results.

To evaluate time alignment intrinsically, we align the sausage with the ground truth transcript by edit distance, and measure the midpoint time stamp error (TSE, in milliseconds) of the correct words. Results on CALLHOME are shown in Figure 1. We present the bias-unshifted TSE distribution for Dev set on the left, and the shifted version on Eval on the right. In general, the time alignments obtained by all three methods are accurate, with average TSE bias less than 12 ms on CALLHOME Eval. Among them, the alignments obtained from the HMM-GMM aligner (`tri3`) trained on ASR training data work the best, while `k2`-based decoder with word-level configuration has the largest bias and variance.

On the other hand, we perform an extrinsic evaluation of time alignment with KWS. We report TWVs with increasing temporal tolerance $\Delta_T = 0.25, 0.5, 5.0$ seconds in Table 4. Among the four methods, aligners `tri3` and `k2`-token work better for KWS across all Δ_T's, due to smaller bias and variance of TSE. Their ATWVs do not increase much when Δ_T increases, even when $\Delta_T = 5.0$s. This means time alignment is not the bottleneck for the KWS performance. In other words, improving the alignment performance on these datasets will not bring further ATWV gain.

5.3. Evaluation of Word Confidence Scores

We evaluate different confidence estimation methods described in Section 4. For the softmax probability, we experiment with acoustic model plus language model scores (AM+LM), attention scores (ATT) and CTC scores (CTC). Performance is measured by normalized cross entropy (NCE), expected calibration error (ECE) and the area under precision-recall curve (AUPR), following e.g., [16, 17] as intrinsic evaluation. We report the metrics across all sausage arcs, just on the 1-best hypothesis, or restricted to only the words in the keywords list. We then report KWS results as extrinsic

[4] `https://www.nist.gov/document/kws16-evalplan-v04.pdf`

[5] `https://github.com/k2-fsa/k2`

[6] `https://github.com/k2-fsa/icefall`

Method	STD2006 Eval				CALLHOME Eval			
	NCE ↑	ECE % ↓	AUPR % ↑	ATWV	NCE ↑	ECE % ↓	AUPR % ↑	ATWV
∗ **POST**	0.70 / -0.73 / 0.77	**0.97** / 5.23 / **0.65**	94.18 / 95.61 / **95.77**	**0.8121**	0.60 / -0.76 / 0.57	1.53 / 5.61 / 1.60	89.92 / 92.65 / 90.60	**0.8031**
AM+LM	0.63 / -0.02 / 0.68	3.93 / 6.51 / 3.41	97.30 / 94.09 / 93.46	0.7795	0.52 / -0.04 / 0.62	5.28 / 9.41 / 3.54	89.08 / 96.10 / 92.03	0.7919
ATT	0.69 / 0.18 / 0.69	2.97 / 4.67 / 3.82	96.20 / 90.09 / 92.73	0.7796	0.59 / 0.21 / 0.67	4.51 / 4.92 / 2.12	83.51 / 94.29 / 88.20	0.7907
CTC	0.66 / -0.02 / 0.73	2.27 / 5.59 / 1.79	**97.41** / 94.43 / 91.81	0.7783	0.61 / 0.03 / 0.61	2.51 / 6.90 / 2.87	89.10 / **96.11** / 91.29	0.7853
CEM	**0.77 / 0.21 / 0.82**	1.67 / **1.83** / 1.61	96.59 / **96.56** / 95.47	0.8079	**0.72 / 0.24 / 0.70**	**1.33 / 2.51 / 1.53**	**91.76** / 94.85 / **92.69**	0.7937

Table 3. Intrinsic and extrinsic evaluation of various confidence scores. For NCE, ECE or AUPR, the metrics are reported "across all sausage arcs / on 1-best / on sausage but restricted to the words in the keywords list". The closer to 1.0 (NCE, AUPR, ATWV), or to 0.0 (ECE), the better. The best number of each column is in bold. The underlined numbers can refer to Table 2.

KWS system	STD2006 Eval	CALLHOME Eval
∗ **E2E, `tri3`**	**0.8040 / 0.8121 / 0.8145**	**0.7884 / 0.8031 / 0.8073**
E2E, MFA	0.7700 / 0.7902 / 0.7983	0.7502 / 0.7725 / 0.7940
E2E, `k2`-token	0.7842 / 0.8042 / 0.8067	0.7780 / 0.7854 / 0.7895
E2E, `k2`-word	0.7153 / 0.7970 / 0.8067	0.7483 / 0.7859 / 0.7883

Table 4. Extrinsic evaluation of alignments: comparing KWS results ($\text{ATWV}_{\Delta_T=0.25}$ / $\text{ATWV}_{\Delta_T=0.5}$ / $\text{ATWV}_{\Delta_T=5.0}$) on Eval sets for different ways to obtain alignments. The smaller the gap between three numbers in each cell, the better.

evaluation for those methods accordingly.

The results are presented in Table 3. Overall, from the column of ATWV for each dataset, all confidence estimation methods for E2E system provide reasonable KWS performance. Yet, they still fall behind the hybrid baseline (row 2 of Table 2). From our experiments, we found all intrinsic metrics, measured either across all sausage arcs or just on the 1-best hypothesis, *do not* correlate well with KWS performance.

For example, in Table 3, the confidence estimated by the dedicated CEM module outperforms the others in NCE, ECE and AUPR in most cases. We expected that the best KWS results would come from the CEM method, but this was *not* the case. Using the CEM resulted in good KWS performance, but the best KWS result was obtained by the method based on the aggregated sequence-level posteriors (POST). This is quite surprising, as POST is the simplest to implement. The discrepancy may be explained by two reasons, one being that the intrinsic metrics have ignored word identities while TWV averages *per-word* performance, the other being that in KWS the ordering of the hits matters more than the absolute values of their confidence scores. This highlights the need to evaluate confidence scores using downstream tasks such as KWS.

5.4. Evaluation of Word Recall

In Table 5 and 6, we compare the word recall of the N-best lists generated by different systems or decoders. We vary the decoding beam size of E2E ASR and the size of the N-best lists. We measure the oracle WER and KWS performance.

We see that while each system has a similar WER (Table 1), the hybrid system has the lowest oracle WER comparing to most of others (Table 5). It also has the highest STWV. While most of the time, the oracle WER correlates well with the ATWV, it is not always the case. We note that when the beam size or the size N has increased to some point, ATWV stops to increase (e.g., the last two rows of Table 5 and 6) or even starts to decrease. This is possibly due to either the lack of decoding diversity of E2E ASR, or the sentence-level posterior-based confidence distribution being flattened (when N is large and temperature scaling is applied), or there being more noises or false alarms included into the N-best lists.

50-best Lists		STD2006 Eval		CALLHOME Eval	
		Oracle WER%	ATWV / STWV	Oracle WER%	ATWV / STWV
Hybrid		6.4	**0.8389 / 0.9247**	8.8	**0.8534 / 0.9136**
E2E	**k2**	6.9	0.7991 / 0.8829	9.8	0.7866 / 0.8627
	Beam 20	6.9	0.8021 / 0.8844	10.3	0.7945 / 0.8567
	∗**Beam 40**	6.3	0.8121 / 0.8972	9.6	0.8031 / 0.8720
	Beam 100	**5.9**	0.8097 / 0.8981	9.2	0.7973 / 0.8695

Table 5. Comparing oracle WER and ATWV/STWV of various ASR systems and various beam sizes for E2E ASR's beam search decoder.Lower WER or higher ATWV/STWV is better.

Size of N-best List	STD2006 Eval		CALLHOME Eval	
	Oracle WER%	ATWV / STWV	Oracle WER%	ATWV / STWV
N=1	12	0.7308 / 0.8092	18.8	0.7274 / 0.7598
N=10	7.3	0.7868 / 0.8544	11.9	0.7859 / 0.8382
∗**N=50**	6.5	**0.8121** / 0.8972	9.6	**0.8031** / 0.8720
N=100	**5.0**	0.8037 / **0.8978**	**8.5**	0.7994 / **0.8736**

Table 6. KWS performance (ATWV/STWV) w.r.t the size of N-best lists. Lower WER or higher ATWV/STWV is better.

6. CONCLUSIONS

In this paper, we built KWS system based on E2E ASR's N-best lists. We explored different ways to recover timing and estimate word-level confidence scores. The KWS performance of E2E ASR still falls behind its hybrid ASR counterpart, even though E2E ASR has better WER. To inspect this, we provide extensive intrinsic and extrinsic evaluations for timing, confidence scores and word recall. We found that the recovered timing is quite accurate and the estimated confidence scores are reasonably good, while the word recall of E2E ASR may be accountable for the KWS performance gap. Future research may consider improving the recall of E2E ASR by methods such as diversified beam search, multi-pass decoding, ASR error correction or contextualized ASR that recalls rare words. Reconciling intrinsic and extrinsic confidence metrics or reducing false alarms for KWS would also be useful.

7. REFERENCES

[1] A. Mandal, KR Prasanna Kumar, and P. Mitra, “Recent developments in spoken term detection: a survey,” *International Journal of Speech Technology*, vol. 17, no. 2, pp. 183–198, 2014.

[2] Gábor Gosztolya, “On the concept of correct hits in spoken term detection,” in *AIC*, 2014.

[3] Jan Trmal, Matthew Wiesner, Vijayaditya Peddinti, Xiaohui Zhang, Pegah Ghahremani, Yiming Wang, Vimal Manohar, Hainan Xu, Daniel Povey, and Sanjeev Khudanpur, “The Kaldi OpenKWS system: Improving low resource keyword search,” in *INTERSPEECH*, 2017.

[4] Andrew Rosenberg, Kartik Audhkhasi, Abhinav Sethy, Bhuvana Ramabhadran, and Michael Picheny, “End-to-end speech recognition and keyword search on low-resource languages,” *ICASSP*, 2017.

[5] Gui-Xin Shi, Weiqiang Zhang, Guan-Bo Wang, Jing Zhao, Shuzhou Chai, and Ze-Yu Zhao, “Timestamp-aligning and keyword-biasing end-to-end ASR front-end for a KWS system,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, pp. 1–14, 2021.

[6] Runyan Yang, Gaofeng Cheng, Haoran Miao, Ta Li, Pengyuan Zhang, and Yonghong Yan, “Keyword search using attention-based end-to-end ASR and frame-synchronous phoneme alignments,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3202–3215, 2021.

[7] Jan Švec, Luboš Šmídl, Josef V. Psutka, and Aleš Pražák, “Spoken term detection and relevance score estimation using dot-product of pronunciation embeddings,” *INTERSPEECH*, 2021.

[8] Gaofeng Cheng, Haoran Miao, Runyan Yang, Keqi Deng, and Yonghong Yan, “ETEH: Unified attention-based end-to-end ASR and KWS architecture,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1360–1373, 2022.

[9] Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury, “End-to-end ASR-free keyword search from speech,” *ICASSP*, pp. 4840–4844, 2017.

[10] Zeyu Zhao and Weiqiang Zhang, “End-to-end keyword search based on attention and energy scorer for low resource languages,” in *INTERSPEECH*, 2020.

[11] Bolaji Yusuf, Alican Gök, Batuhan Gündogdu, and Murat Saraçlar, “End-to-end open vocabulary keyword search,” *INTERSPEECH*, 2021.

[12] Jinyu Li, “Recent advances in end-to-end automatic speech recognition,” *ArXiv*, vol. abs/2111.01690, 2021.

[13] Ye Bai, Jiangyan Yi, Hao Ni, Zhengqi Wen, Bin Liu, Ya Li, and Jianhua Tao, “End-to-end keywords spotting based on connectionist temporal classification for mandarin,” *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 1–5, 2016.

[14] Tara N. Sainath, Ruoming Pang, David Rybach, Basi García, and Trevor Strohman, “Emitting word timings with end-to-end models,” in *INTERSPEECH*, 2020.

[15] Rui Zhao, Jian Xue, Jinyu Li, Wenning Wei, Lei He, and Yifan Gong, “On addressing practical challenges for RNN-transducer,” *ASRU*, pp. 526–533, 2021.

[16] Alexandros Kastanos, Anton Ragni, and Mark John Francis Gales, “Confidence estimation for black box automatic speech recognition systems using lattice recurrent neural networks,” *ICASSP*, pp. 6329–6333, 2020.

[17] Mingqiu Wang, Hagen Soltau, Laurent El Shafey, and Izhak Shafran, “Word-level confidence estimation for RNN transducers,” *ASRU*, pp. 1170–1177, 2021.

[18] Lidia Mangu, Eric Brill, and Andreas Stolcke, “Finding consensus among words: lattice-based word error minimization.,” in *EUROSPEECH*. Citeseer, 1999.

[19] Dogan Can and Murat Saraçlar, “Lattice indexing for spoken term detection,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 2338–2347, 2011.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., “The Kaldi speech recognition toolkit,” in *ASRU*, 2011.

[21] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using Kaldi,” in *INTERSPEECH*, 2017.

[22] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, 2006.

[23] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.

[24] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *ICASSP*, 2017.

[25] Yajie Miao, Mohammad Gowayyed, and Florian Metze, “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in *ASRU*, 2015.

[26] Aleksandr Laptev, Somshubra Majumdar, and Boris Ginsburg, “CTC variations through new WFST topologies,” *arXiv preprint arXiv:2110.03098*, 2021.

[27] Frank Wessel, Klaus Macherey, and Ralf Schlüter, “Using word probabilities as confidence measures,” *ICASSP*, pp. 225–228 vol.1, 1998.

[28] Dan Oneata, Alexandru Caranica, Adriana Stan, and Horia Cucu, “An evaluation of word-level confidence estimation for end-to-end automatic speech recognition,” *SLT*, 2021.

[29] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, “Enriching word vectors with subword information,” *ACL*, vol. 5, pp. 135–146, 2017.

[30] Jonathan G Fiscus, Jerome Ajot, John S Garofolo, and George Doddingtion, “Results of the 2006 spoken term detection evaluation,” in *Proc. SIGIR*, 2007, vol. 7, pp. 51–57.

[31] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Yalta, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, “ESPnet: End-to-end speech processing toolkit,” *INTERSPEECH*, 2018.