END-TO-END LOW RESOURCE KEYWORD SPOTTING THROUGH CHARACTER RECOGNITION AND BEAM-SEARCH RE-SCORING

Ephrem Tibebe Mekonnen[1], *Alessio Brutti*[2], *Daniele Falavigna*[2]

(1) University of Trento (Italy),
(2) Fondazione Bruno Kessler, Trento (Italy)
{brutti,falavi}@fbk.eu,ephrem.mekonnen@studenti.unitn.it

ABSTRACT

This paper describes an end-to-end approach to perform keyword spotting with a pre-trained acoustic model that uses recurrent neural networks and connectionist temporal classification loss. Our approach is specifically designed for low-resource keyword spotting tasks where extremely small amounts of in-domain data are available to train the system. The pre-trained model, largely used in ASR tasks, is fine-tuned on in-domain audio recordings. In inference the model output is matched against the set of predefined keywords using a beam-search re-scoring based on the edit distance.

We demonstrate that this approach significantly outperforms the best state-of-the art systems on a well known keyword spotting benchmark, namely "google speech commands". Moreover, compared against state-of-the-art methods, our proposed approach is extremely robust in case of limited in domain training material. We show that a very small performance reduction is observed when fine tuning with a very small fraction (around 5%) of the training set.

We report an extensive set of experiments on two keyword spotting tasks, varying training sizes and correlating keyword classification accuracy with character error rates provided by the system. We also report an ablation study to assess on the contribution of the out-of-domain pre-training and of the beam-search re-scoring.

Index Terms— keyword spotting, beam search, connectionist temporal classification, low resources, end-to-end recognition

1. INTRODUCTION

The problems of keyword spotting (KWS) and spoken term detection (STD) have been largely investigated in the past by the scientific community, often employing approaches borrowed from automatic speech recognition (ASR).

Concerning STD, several challenges [1, 2, 3] have been organised by NIST and data-sets, also collected in very noisy conditions [4, 5], have been released in order to produce reference benchmarks. More recently, the scientific community can benefit from public data-sets [6, 7, 8] specifically designed for investigating KWS tasks. In particular, the Google Speech Command (*GSC*) dataset [8] has been experimented by many labs and hence can be considered the current de-facto benchmark for KWS. Recently, very efficient solutions has been developed leading to extremely high state-of-the-art accuracy.

Large vocabulary speech recognition (LVSR) systems were often employed as a flexible way to solve KWS tasks. Their main advantage is that they do not require training specific keywords models depending on the particular application. However, LVSR systems need very heavy models to be effective, making them eager of computational and memory resources.

As an alternative, keywords models can be built by concatenating their corresponding phone models [9, 10] so that they act as fillers of the keywords to spot. This approach allows to solve open vocabulary tasks, where the keyword dictionary can change without requiring to train application specific models.

A research area closely related to KWS, that has received great attention in the last years by the scientific community is "query by examples" [11, 12]. It aims at finding occurrences of a spoken query inside audio streams. This technology has become increasingly interesting in the last years thanks to the ability of searching and indexing audio documents even when the languages, both of the query and of the database, are unknown or when low resources are available for training models. Actually, recent approaches make use of embedded representations [12] of audio signals so that there is no need to train language dependent acoustic models. For KWS, however we cannot assume to always have available spoken instances of the keywords. Conversely, the requirement is often to "freely" search a limited number of given keywords inside audio streams.

In this paper we propose a KWS approach, suitable for open vocabulary and low-resource applications, based on the recurrent neural network (RNN) model described in [13]. The model implements bidirectional long-short-term memory (LSTM) cells and outputs symbols corresponding to: characters of an alphabet (English in this work), space (end-of-word) and blank. The model is trained by maximising a connectionist temporal classification (CTC) loss [13]. During decoding it produces a list of character hypotheses that best match the input speech, which are then re-scored against the list of predefined keywords. Finally, the most likely keyword is selected as output.

Note that this approach is substantially different from those, previously mentioned, based on filler models [9, 10], since it doesn't require to build models of keywords based on their phonetic transcriptions. Instead, the keywords are searched along the string of characters yielded by the decoder. This implies to investigate: *a)* accurate and fast methods to generate the character hypotheses, and *b)* effective approaches to detect the keywords inside the decoder output. Although the scientific community has just started to investigate along this line (some related papers are summarised in section 2), we believe that there is still room for further improvements. Along this line of research, [14, 15] suggest methods for fast keyword search over frame level CTC hypotheses. Conversely, we suggest to perform keyword search over the word boundaries hypothesised by the CTC decoder.

To summarzie, our contribution to the KWS task is as follows.

- We propose to use a beam-search decoder working on the output of a CTC-trained neural model.

This work has been partially funded by the European Institute of Innovation Technology (EIT) within the SmarTerp Project under contract n. 21184.

- We pre-train the CTC model on a large out-of-domain data-set, namely *LibriSpeech* [16], and fine tune the model on different quantities of in-domain data extracted from the *GSC* training corpus. In this way, the model outperforms, to the best of our knowledge, the state-of-the-art on the *GSC* data-set, and provides high accuracy even if a small dataset (i.e. less than one hour) is employed for fine tuning.
- We propose a method (see section 3) for searching the keyword that maximises the joint probability of an hypothesis in the beam and the keyword itself, given an input sequence of acoustic observations.

Finally, we point out that this work is preparatory to the development of a system capable of handling real time, open vocabulary KWS applications on continuous audio streams.

2. RELATED WORKS

With the extraordinary improvements of acoustic models brought by deep neural networks, in particular RNNs [17, 13, 18, 19] and Transformers [20, 21], architectures for open keywords spotting have been developed and tested on several data-sets, both private [22, 14] and public, such as the previously mentioned *GSC* corpus [23, 24, 25, 26]. In these approaches, similarly to what we are proposing, the neural models are employed to predict the characters in the input speech. For this task, a fair comparisons between RNN-CTC models and Transformers can be found in the paper from Facebook [21].

The work described in [27] proposes to use a keyword network acting as filler model of the keywords to detect. This network processes the output of a RNN-CTC model that provides, in addition to the sequence of characters, its corresponding posterior probability (this information is also used by the keyword search strategy used in this paper). A RNN-CTC model is also employed in [28], where it is demonstrated that it outperforms a hybrid DNN-HMM model for KWS on a proprietary Mandarin data set. In a similar way, the work described in [29] proposes an end-to-end architecture for both KWS and voice activity detection. This CTC model was trained on a large data-set collected from an android phones assistant application and, also in this case, it outperforms a hybrid DNN-HMM model. More recently, the work described in [22] proposes to append a convolutional neural network, that processes the spectrogram of the input signal, at the bottom of a RNN-CTC model. This idea has demonstrated effective for KWS on a clean Mandarin speech database, using as output units: tonal syllables, characters or whole keywords.

LSTM with CTC has been also applied in a spoken language understanding task [14]. The authors of this paper propose a fast detection algorithm for keyword search, at frame level. Since this last operation is extremely expensive from a computational point of view then, in order to avoid to perform a full search over the input audio, a very recent paper [15] suggests to utilise the attention mechanism in Transformer networks to predict keywords endpoints. Then, reference keywords and decoded character sequences inside endpoints are compared as in [14].

The main difference between the approaches mentioned above and the one proposed in this paper lies in the method employed to score the hypotheses in the beam, as explained in section 3.2, as well as in the usage of an out-of-domain pre-trained model.

3. PROPOSED APPROACH

In this section we describe our keyword spotting approach based on end-to-end character recognition, in particular for what concerns the keyword search and the related beam-search re-scoring strategy.

3.1. Character-level connectionist temporal classification

CTC [13, 18, 19] is a sequence-to-sequence approach for sequence labelling tasks, that allows to predict labels and "blank" symbols at any time of the input. The capability of predicting "blank" (i.e. no character as output) allows the model to produce output sequences of different lengths from the input. Although CTC loss has been initially investigated as a sequence-to-sequence learning algorithm for RNNs, it is largely used also in combination with cross-entropy loss in modern ASR architectures (e.g. Transformers [20, 21]) .

Let us consider the set of all possible character symbols $\mathcal{D} = \emptyset \bigcup \{d_1, \ldots, d_L\}$, where L is the set dimensionality and $\emptyset$ is the blank symbol. We define as $\mathcal{B}$ a "many-to-one" aligning operator that maps a given input feature sequence $\mathbf{x} = x_1, \ldots, x_T$ of length T, into all possible target character sequence $\mathbf{s}$, obtained considering possible repetitions of each symbol $d_l \in \mathcal{D}$. The posterior probability $p[\mathbf{s}|\mathbf{x}]$ of the character sequence given the input features can be computed as:

$$p[\mathbf{s}|\mathbf{x}] = \sum_{\pi \in \mathcal{B}^{-1}} p[\pi|\mathbf{x}] \tag{1}$$

Basically, $p[\mathbf{s}|\mathbf{x}]$ is the sum of posterior probabilities of all possible paths $\pi \in \mathcal{B}^{-1}$ that align $\mathbf{s}$ with $\mathbf{x}$, and can be computed with the forward-backward algorithm [13]. Finally the CTC loss is defined as:

$$L_{CTC}(\mathbf{s}) = -\log(p[\mathbf{s} \mid \mathbf{x}]) \tag{2}$$

At inference, the CTC decoder evaluates at each time step t the probabilities $p[d_l|x_t], d_l \in \mathcal{D}$, and accumulates them along the partial paths $\Pi^k_{1:t} (1 \leq k \leq K)$ contained in a ordered stack of hypotheses (i.e. a beam search strategy is applied) of size K (in doing this the algorithm distinguishes between blank and non-blank probabilities). At the end, the beam contains the best K aligned hypotheses $\Pi^k = \Pi^k_{1:T}$ with their associated posterior probabilities $P[\Pi^k|\mathbf{x}]$.

3.2. Keyword search

The most straightforward approach for finding the "best" keyword along the sequence of characters furnished by the CTC decoder is picking the keyword w^* that minimises the edit distance between the best hypothesis in the beam, i.e. Π^1, and the words in the lexicon $\mathcal{L}$, that is:

$$w^* = \underset{w \in \mathcal{L}}{\texttt{argmin}} \left(edit(w, \Pi^1) \right) \tag{3}$$

where $edit(\cdot)$ is the edit distance between two character sequences. However equation 3, considering only 1 hypothesis ($K = 1$), does not take into account all acoustic information embedded in the input and does not consider the set of possible keywords during sequence decoding. Therefore, we derive a more formal approach to keyword search that also takes explicitly into account the acoustic information in the input $\mathbf{x}$. We suggest to look for the keyword w^* in the lexicon $\mathcal{L}$ that maximises the joint log-probability $\log \left(P[\Pi^k, w|\mathbf{x}] \right)$. That is:

$$w^* = \underset{w \in \mathcal{L}, k=1:K}{\texttt{argmax}} \log \left(P[\Pi^k, w|\mathbf{x}] \right) \tag{4}$$

Applying the product Bayes formula, the joint log-probability can be rewritten as:

$$\log \left(P[w, \Pi^k|\mathbf{x}] \right) = \log \left(P[\Pi^k|\mathbf{x}] \right) + \log \left(P[w|\Pi^k, \mathbf{x}] \right) \tag{5}$$

In the equation above, the search is carried out over the whole set of possible keywords $w \in \mathcal{L}$ and for all K hypotheses in the beam

($k = 1 : K$). According to [30], we can interpret the normalised edit distance as the error probability of a sequence of characters given the reference string (when the different edit operations have the same weights we can assume that the maximum probability path corresponds to the one with the minimum edit distance). Therefore, we can approximate the posterior probability of a keyword w given an hypothesis Π^k as follows:

$$\hat{P}[w \mid \Pi^k, \mathbf{x}] = \left(1 - \frac{edit(w, \Pi^k)}{len(w)}\right) \quad (6)$$

Although the normalisation term, according to [30], should be equal to the length of the best alignment path here, for the sake of simplicity, we use the length of the keyword $len(w)$.

Finally, the best keyword w^* can be computed following equation 5 and introducing a parameter α that weighs differently the contribution of each of the two terms in the summation, that is:

$$w^* = \underset{w \in \mathcal{L}, k=1:K}{\texttt{argmax}} \; \alpha \log(P[\Pi^k|\mathbf{x}]) + (1-\alpha)\log(\hat{P}[w|\Pi^k, \mathbf{x}]) \quad (7)$$

The introduction of the parameter α is due to the approximated nature of the probabilities in the equation above. Note that if we don't consider the normalisation term "$len(w)$" in equation 6, then equation 7 reduces to equation 3 for $\alpha = 0$ and $K = 1$.

It is worth observing that although the data employed in this work (see section 4) contains the exact time boundaries of the keywords, this doesn't represent a limitation to the application of the described approach to a continuous audio stream, since we can apply it to each character sequence surrounded by the word-end labels provided by the CTC decoder.

4. EXPERIMENTAL ANALYSIS

As mentioned above, we evaluate our proposed approach on the *GSC* data-set [8]. The Version 2 of the data-set includes 35 different English words uttered by various speakers for a total of 105K recordings. Each recording is 1 second long and it is sampled at 16kHz. Following the best practice in literature, we evaluate our proposed approach considering 2 recognition tasks.

- V2-35: recognition of all 35 words using *GSC* Version 2
- V2-12: recognition of 10 keywords ("Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", and "Go") plus "Unknown" and "Silence" using V2.

For both tasks we use the official partitions of the data-set.

We compare our performance with the current best performing state-of-the-art (SOTA) methods according to the *GSC* leaderboard[1]. In particular, we consider: *a)* Audio Spectrogram Transformer (AST2021) [31] the first purely attention-based model which outperforms other KWS approaches on V2-35 task, *b)* Broadcasted Residual Learning (BC-ResNet8) [23] achieves competitive accuracy with a small model size and computation load, *c)* Res15 [26] learns efficient representations for KWS with triplet loss-based metric embeddings, *d)* Keyword Transfomer (KWT-1-2-3) [24] use Transformer architecture for KWS and *f)* the attention RNN model in [25] which is not among the top performing but is the current best recurrent model and, hence, it represents a good baseline for our architecture. The proposed "CTC-RNN + librispeech + eq. 7" achieves best STOA accuracy on V2-12 task.

[1]https://paperswithcode.com/sota/keyword-spotting-on-google-speech-commands

4.1. Implementation details

The network employs bidirectional LSTM (BLSTM) cells and takes as input 80 mel frequency cepstral coefficients (MFCCs) feature vectors extracted from the input signal using an analysis window of 20ms at 10ms frame step. They are computed from a bank of 80 filters and no context is added. The network has three layers of 250 BLSTM cells with linear activation function. The output layer consists of as many logits as the possible character symbols plus the blank and special ones, and implements softmax as activation. Overall, the output labels are 29 in total: 26 correspond to English alphabet characters, one to the special character "‘", one to the word boundary label ">" and one to the blank label "∅". Target labels correspond to the characters of the training utterance transcriptions with the label ">" inserted at the end of each word.

The network was trained with stochastic gradient descent and ADAM algorithm [32], starting with a learning rate equal to 10^{-5} which is reduced along the training epochs.

The batch size was set to 32 for pre-training on *LibriSpeech* and to 16 for both training and fine tuning on the *GSC* data-set. A dropout factor equal to 0.5 was used for regularisation in all layers. The total number of parameters of the network is 1,343,000. The model is pre-trained using 1000 hours from *LibriSpeech* [16] with the same hyperparameters mentioned above. The value of the beam size was empirically fixed to K=10. Finally, α has been optimised in each experiment on the *GSC* validation set.

The whole pytorch code of the system is available at `https://github.com/Ephrem-ETH/E2E-KWS`.

4.2. Results

Table 1 compares the performance of the selected SOTA methods against our proposed approach. In particular, the table reports also an ablation study that highlight the contribution of equation 7 with respect to equation 3 as well as the impact of the model pre-trained on *LibriSpeech*. Results of the comparative methods are taken from the related papers.

Table 1. Results of our model as compared to STOA models.

Model	**Accuracy (%)**	
	V2-12	***V2-35***
AST	-	**98.11**
BC-ResNet8	98.70	-
Res15	98.56	97.00
KWT-3	98.56	97.69
KWT-2	98.43	97.74
KWT-1	98.08	96.95
Attention RNN	96.90	93.90
CTC-RNN + equation 3	94.18	84.00
CTC-RNN + equation 7	95.07	85.92
CTC-RNN + librispeech + eq. 3	98.62	95.06
CTC-RNN + librispeech + eq. 7	**98.95**	95.85

Considering the V2-12 task, our method performs better than any other SOTA approach, in particular the other recurrent model. Given the small size of the *GSC* dataset, pre-training the model using a large data-set, even if out of domain, brings a very large improvement which ranges between 3 and 4 percent points. Note also that equation 7 allows to achieve almost 1 percent improvement without pre-training. This contribution is smaller with pre-training because the accuracy is almost saturated, but still evident.

Considering the task V2-35, however, our method fails short of the current best approaches, although it performs considerably better

than the other recurrent model. This can be attributed to the fact that, as expected, the KWS accuracy strongly depends on the performance of the character classifier. This is shown in Figure 1 that reports the KWS accuracy as a function of the character error rate (CER) obtained with the different models (trained using different amount data as explained later). Note the large difference in accuracy between

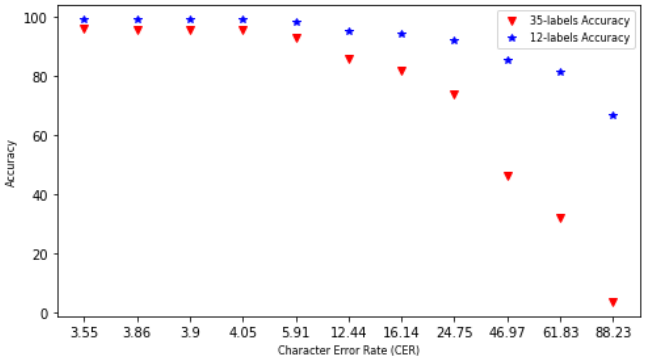

Fig. 1. Accuracy of the KWS system as a function of the CER of the character recogniser.

the V2-35 and V2-12 tasks for high values of CER, corresponding to the weakest acoustic models. We observed that the worst models, trained with only 5% of *GSC* data, almost always outputs empty strings which are mapped, through equation 7, into a same word (i.e.: "go"). In this way, while many out-of-dictionary words in the V2-12 task are correctly rejected, in the V2-35 task they are mis-recognised and counted as errors. To conclude, the ablation study for the task V2-35 confirms the substantial improvements provided by the pre-training stage, with in this case is a much as 10 percent points. The beam-search re-scoring strategy, also in this case, provides almost 1 percent point.

4.3. Low Resource keyword spotting

We conclude the experimental analysis, evaluating the behaviour of our proposed approach when limited training material is available, which is the main purpose behind the design of our method. To this end, Figure 2 reports the keyword classification accuracy obtained with different fractions (from 5% to 100%) of the *GSC* training data, on both recognition taks, with and without pre-training. It is interesting to observe that on the V2-12 task, when pre-training the model on *LibriSpeech*, 25% of the training data is sufficient to almost reach state of the art performance. In addition, a very mild degradation in performance is observed with as less as 5% of the *GSC* training material (i.e. less than one hour). A similar behaviour is observed in the V2-35 tasks with less than one percent point reduction when using only 25% of the training set. On the contrary large reduction in the accuracy is observed without pre-training. This result is in line with many others reported in the scientific literature. As an example, the pre-trained "wav2vect" model [33, 34, 35] outperforms, after fine tuning over one hour of *LibriSpeech*, a model trained on 100 hours of the same database.

Figure 3 shows the performance difference between the application of equations 3 and 7 for keyword search. Also in this case KWS accuracy is plotted against the fraction of *GSC* material used to train the neural model (for reasons of graphic clarity we report only accuracy for the V2-35 task). We can observe the significant contribution

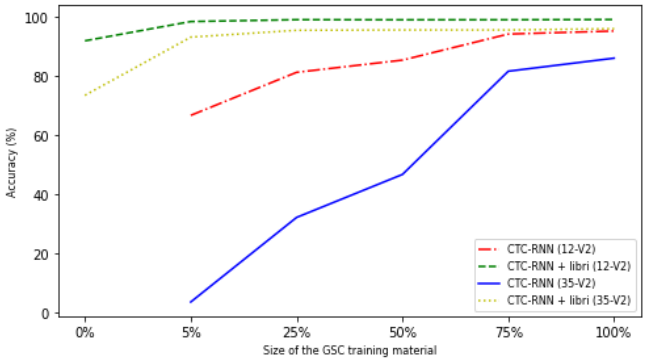

Fig. 2. KWS accuracy as a function of different amounts of *GSC* training material and for two tasks. Models were either trained from scratch or fine tuned from a model pre-trained on *LibriSpeech*.

of the proposed Bayesian formulation for beam hypotheses rescoring, especially at the lowest levels of KWS accuracy.

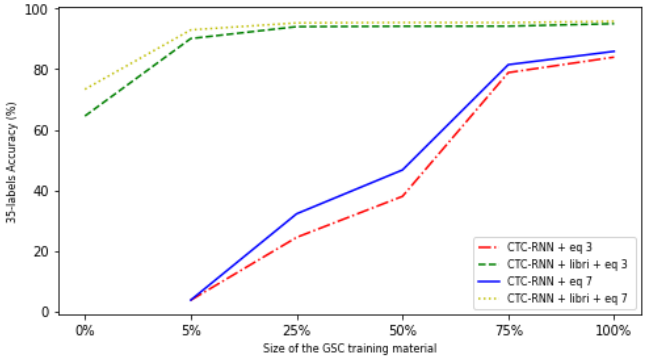

Fig. 3. Comparison between the two proposed keyword search approaches (see eq. 3 and eq. 7). Models were either trained from scratch or pre-trained on *LibriSpeech*. Task V2-35.

5. CONCLUSION AND FUTURE WORK

In this paper we described an end-to-end system for low-resource keyword spotting and evaluated in on the Google Speech Command corpus, extensively used as benchmark by the scientific community. The proposed method uses a neural models formed by 3 layers of bidirectional LSTM cells and is trained by optimising a CTC loss. We have demonstrated that a model pre-trained on 1000 hours of the out-of-domain *LibriSpeech* dataset and successively fine tuned on the *GSC* training corpus significantly outperforms a model trained only on the *GSC* corpus. In addition, a small performance decrease has been observed even if fine tuning is carried out on a small fraction (less than one hour) of the *GSC* training data. Finally, we have described a Bayesian framework for keyword search and shown its effectiveness. Future works will address more in depth comparisons with SOTA approaches using pre-trained models (e.g. by employing some of the best models reported in the *LibriSpeech* leaderboard) , as well as the application of the proposed Bayesian search to other data sets and KWS challenging tasks, in particular the one described in [36], where the "most important" words of a talk have to be detected in real time.

6. REFERENCES

[1] J. Fiscus, J. Ajot, J. Garafolo, , and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. of ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, Amsterdam, Holland, 2007.

[2] "Intelligence advanced research projects activity (iarpa), babel program," URL:https://www.iarpa.gov/index.php/research-programs/babel.

[3] "National institute of standards and technology (nist), openkws13 keyword search evaluation plan," March 2013, URL:https://www.nist.gov/system/files/documents/itl/iad/mig/OpenKWS13-EvalPlan.pdf.

[4] K. Walker and S. Strassel, "The rats radio traffic collection system," in *Proc. of ISCA Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012.

[5] L. Mangu, H. Soltau, H-K. Kuo, and G. Saon, "The ibm keyword search system for the darpa rats program," in *Proc. of ASRU*, 2012, pp. 204–209.

[6] A. Ghandoura, F. Hjabob, and O. Al Dakkak, "Building and benchmarking an arabic speech commands dataset for small-footprint keyword spotting," *Engineering Applications of Artificial Intelligence*, vol. 102, 2021.

[7] B. Kim, M. Lee, J. Lee, Y. Kim, and K. Hwang, "Query-by-example on-device keyword spotting," in *Proc. of ASRU*, Sentosa, Singapore, 2019.

[8] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, vol. abs/1804.03209, 2018, URL:http://arxiv.org/abs/1804.03209.

[9] R.C. Rose and D.B. Paul, "A hidden markov model based keyword recognition system," in *Proc. of ICASSP*, 1990, pp. 129–132.

[10] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, and J. Černocký, "Phoneme based acoustics keyword spotting in informal continuous speech," in *Proc. of TSD*, Czech Republic, 2005, pp. 302–309.

[11] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "High-performance query-by-example spoken term detection," in *Proc. of ICASSP*, 2014.

[12] D. Ram, L. Miculicich, and H. Bourlard, "Neural network based end-to-end query by example spoken term detection," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 28, pp. 1416–1427, 2020.

[13] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. of* 29^{th} *International Conference on Machine Lerning*, 2012, URL:https://arxiv.org/pdf/1211.3711.pdf.

[14] T. Bluche, M. Primet, and T. Gisselbrecht, "Small-footprint open-vocabulary keyword spotting with quantized LSTM networks," *CoRR*, vol. abs/2002.10851, 2020, URL:https://arxiv.org/abs/2002.10851.

[15] B. Wei, M. Yang, T. Zhang, X. Tang, X. Huang, K. Kim, J. Lee, K. Cho, and S. Park, "End-to-end transformer-based open-vocabulary keyword spotting with location-guided local attention," in *Proc. of Interspeech*, 2021, URL:https://arxiv.org/abs/2104.00769.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. of ICASSP*, 2015, pp. 5206–5210.

[17] S. Fernandez, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Artificial Neural Networks – ICANN*, 2007, pp. 220––229.

[18] A. Graves, , Mohamed Abdel-rahman, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of ICASSP*, 2013, pp. 6645–6649.

[19] A. Graves and N. Jaitly, "Towards End-To-End Speech Recognition with Recurrent Neural Networks," in *Proc. of ICASSP*, 2014.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of 31th International Conference on NIPS*, 2017, pp. 6000–6010.

[21] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end ASR: from supervised to semi-supervised learning with modern architectures," *CoRR*, vol. abs/1911.08460, 2019, URL:http://arxiv.org/abs/1911.08460.

[22] H. Yan, Q. He, and W. Xie, "Crnn-ctc based mandarin keywords spotting," in *Proc. of ICASSP*, 2020, pp. 7489–93.

[23] B. Kim, S. Chang, J. Lee, and D. Sung, "Broadcasted residual learning for efficient keyword spotting," in *Proc. of Interspeech*, 2021.

[24] Axel Berg, Mark O'Connor, and Miguel Tairum Cruz, "Keyword transformer: A self-attention model for keyword spotting," in *Proc. of Interspeech*, 2021.

[25] D. Coimbra de Andrade, S. Leo, M. Loesener Da Silva Viana, and C. Bernkopf, "A neural attention model for speech command recognition," *CoRR*, vol. abs/1808.08929, 2018, URL:https://arxiv.org/abs/1808.08929.

[26] R. Vygon and N. Mikhaylovskiy, "Learning efficient representations for keyword spotting with triplet loss," *CoRR*, vol. abs/2101.04792, 2021, URL:https://arxiv.org/abs/2101.04792.

[27] K. Hwang, M. Lee, and W. Sung, "Online keyword spotting with a character-level recurrent neural network," *CoRR*, vol. abs/1512.08903, 2015, URL:http://arxiv.org/abs/1512.08903.

[28] Y. Bai, J. Yi, H. Ni, Z. Wen, B. Liu, Y. Li, and J. Tao, "End-to-end keywords spotting based on connectionist temporal classification for mandarin," in *Proc. of* 10^{th} *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2016, pp. 1–5.

[29] C. T. Lengerich and A. Y. Hannun, "An end-to-end architecture for keyword spotting and voice activity detection," in *NIPS End-to-End Learning for Speech and Audio Processing Workshop*, 2016, URL:http://arxiv.org/abs/1611.09405.

[30] R. Myers, C. Wilson, and E.R. Hancock, "Bayesian graph edit distance," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 22, no. 6, pp. 628–635, 2000.

[31] Y. Gong, Y.-A. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *Proc. of Interspeech*, 2021.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2017, URL:https://arxiv.org/abs/1412.6980.

[33] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. of Interspeech*, 2019.

[34] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. of International Conference on Learning Representations*, 2020, URL:https://arxiv.org/pdf/1910.05453.pdf.

[35] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for asr," in *Proc. of ICASSP*, 2020, pp. 7694–7698.

[36] R. Gretter, M. Matassoni, and D. Falavigna, "Seed words based data selection for language model adaptation," in *Proc. of 18-th MT-Summit*, 2021, URL:https://aclanthology.org/2021.mtsummit-asltrw.1.pdf.