End to End Keyword Spotting Using A Character-Level Recognition and Beam-Search Re-Scoring

Ephrem Mekonnen

University of Trento

Supervisor:
Prof. Elisa Ricci

Co-Supervisors:
Dr. Daniele Falavigna (FBK)
Dr. Alessio Brutti (FBK)

December 21, 2021

Outline

Introduction

Keyword Spotting is a task of detecting keywords of interest in an audio stream.

Few applications of Keyword spotting:

- Awaking voice assistants
- Voice Commands
- Phone call routing
- Detecting sensitive words to find crimes

Motivation

- End-to-End architecture has greatly simplified the pipeline for building and applying the KWS system.
- End-to-end KWS systems have shown to surpass the performance of traditional hybrid DNN-HMM solutions.
- It is challenging to minimize errors while operating efficiently in devices with limited resources such as micro-controllers.

This work:

- proposed a Connectionist Temporal Classification based RNN keyword spotting to tackle these issues.

Proposed Approach

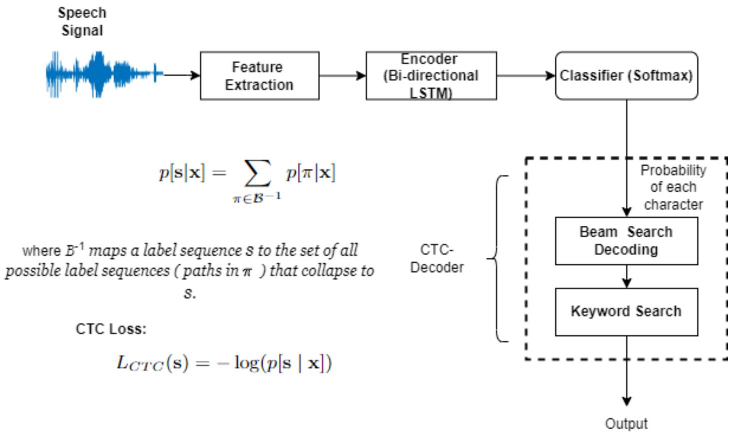

Figure: Architecture of the proposed keyword spotting system

CTC-Decoder: Beam Search Decoder

- Detects the character-level output sequence and gives a probability to each detected character.
- Iteratively creates a stack of partial character hypotheses.
- At each time step, retain only K best scoring partial hypotheses of the previous time step, where K specifies the beam width.

Keyword Search Approaches

Approach 1

To pick 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 $\mathcal{L}$

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

where $edit(\cdot)$ is the edit distance between two character sequences.

Approach 2

The search is carried out over the whole set of possible keywords $w \in \mathcal{L}$ and for all K hypotheses in the beam $(k = 1 : K)$

$$w^* = \underset{w \in \mathcal{L}, k=1:K}{\texttt{argmax}} \ \alpha \log(P[\Pi^k | \mathbf{x}]) + (1 - \alpha) \log(\hat{P}[w | \Pi^k, \mathbf{x}]) \tag{2}$$

where $\hat{P}[w \mid \Pi^k, \mathbf{x}]$ is the posterior probability of a keyword w given an hypothesis Π^k and the acoustic input x.

Training Procedure

- Train the model using 1000 hours of *Librispeech*.
- Train the model from scratch by varying the size of Google Speech Commands (GSC) (version 2) training material.
- Retrain the *Librispeech* trained model by varying the size of *GSC (V2)* training material.
- Evaluate all trained models on test set of *GSC (V2)*.

Evaluation Tasks

- 12-commands recognition task (12-V2)
    - Recognition of 10 keywords ("Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", and "Go") plus "Unknown" and "Silence" using Google Speech Commands(GSC) (Version 2) data set.
- 35-commands recognition task (35-V2)
    - Recognition of all 35 words using *GSC* Version 2

Results

Table: KWS accuracy considering different amounts of *GSC* training material for two tasks. Models were either trained from scratch or fine-tuned from a model pre-trained on *LibriSpeech*.

α	Task	Model	**Accuracy (%)**					
			0% train	5% train	25% train	50% train	75% train	100% train
0.0	12-V2	*GSC* only		66.60	77.86	81.82	92.73	93.53
α^{opt}		*GSC* only		66.62	81.14	85.26	94.06	94.61
α^{opt}		pre-trained	91.77	98.27	98.90	98.86	98.87	98.95
		BC-ResNet8						98.70
0.0	35-V2	*GSC* only		3.63	24.40	37.96	78.89	82.22
α^{opt}		*GSC* only		3.66	32.19	46.69	81.51	85.47
α^{opt}		pre-trained	73.38	93.01	95.32	95.44	95.43	95.85
		AST						98.11

Comparison with STOA Models

Table: 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 $\pm$ 0.07	97.69 $\pm$ 0.09
KWT-2	98.43 $\pm$ 0.08	97.74 $\pm$ 0.03
KWT-1	98.08 $\pm$ 0.10	96.95 $\pm$ 0.14
Attention RNN	96.90	93.90
CTC-RNN eq. 1	94.18	84.00
CTC-RNN eq. 2	95.07	85.92
CTC-RNN+libri eq. 1	98.62	95.06
CTC-RNN+libri eq. 2	**98.95**	95.85

CER Vs Accuracy of KWS

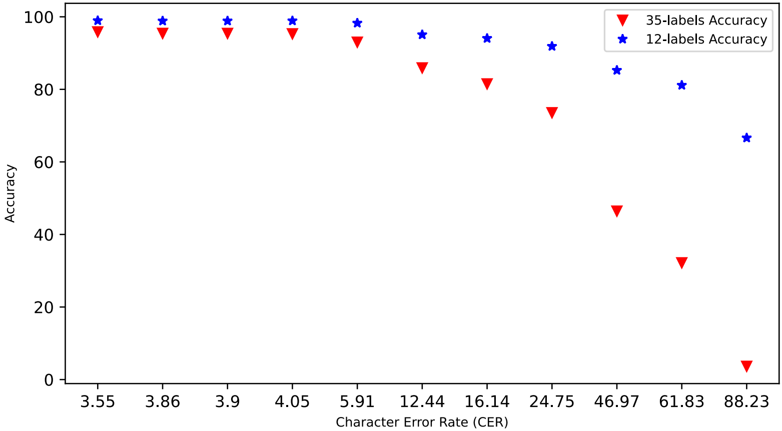

Figure: Accuracy of the KWS system as a function of the CER of the character recogniser.

Conclusion

- Proposes to use a beam search algorithm working on top of the neural network outputs.
- Proposes to use a large out-of-domain dataset to pretrain the CTC model and fine-tuned using less amount of in-domain training data.
- Proposes a new keyword scoring function.

Future Works

- Address more challenging kws task, like detecting the "most important" words of the talk in real time.
- To address such a task we need to improve the performance of the acoustic model in terms of CER, e.g. to use transformer network to improve efficiency.

Thank You