Unified Speculation, Detection, And, Verification Keyword Spotting

amazon alexa

Gengshen Fu, Thibaud Sénéchal, Aaron Challenner, Tao Zhang

gengshef@, thibauds@, aaronlmc@, taozhng@amazon.com

Background & Motivation

- Accurate and timely recognition of the trigger keyword is vital.
- There is a trade-off needed between accuracy and latency.
- Existing works focus on accuracy and computational latency.

Proposed system:

Unified speculation, detection, and verification model

- Speculation makes an early decision, which can be used to give a head-start to downstream processes on the device.
- Detection mimics the traditional keyword trigger task and gives a more accurate decision by observing the full keyword context.
- Verification verifies previous decision by observing even more audio after the keyword span.

Model architecture and training strategy

- Convolutional recurrent neural network (CRNN) architecture
- multi-task learning with different target latencies on the new proposed latency-aware max-pooling loss.

Model Architecture

CNN encoder behaves as an efficient feature extractor to model local temporal and spectral dependencies:

- Convolutional neural network front-end has a receptive field of 34 frames and has a stride of 6 frames.
- Each layer is composed of a convolution layer, a rectified linear unit activation layer, an optional max pooling layer, a batch normalization layer and a drop out layer.
- Outputs are vectorized and fed to RNN decoder

RNN decoder captures dependencies among different frames:

- A long short-term memory (LSTM) layer captures dependencies using “gating” mechanism.
- A full-connected (FC) layer is used to further transform features before Softmax output.
- Due to the similarity of speculation, detection, and verification tasks, i.e., all of them try to detect the same word from audio, we share the same convolutional front-end, LSTM, and FC layer for them to reduce model size.
- We only add three output heads with separate linear layers for dimension reduction and Softmax outputs.
- The additional computations to achieve these three tasks simultaneously are only introduced by these small output heads, hence are negligible.

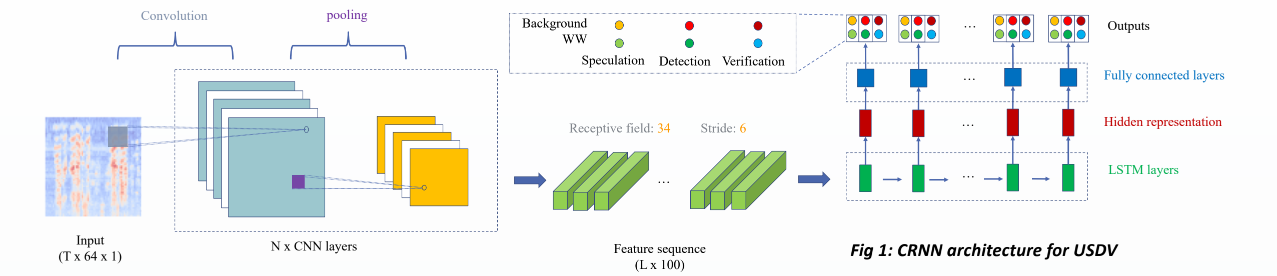

Fig 1: CRNN architecture for USDV

Latency-aware max-pooling loss

- Calculate cross-entropy loss on the frame that has the maximum score on the corresponding class for positive examples.
- Calculate cross-entropy loss on the frame that has the lowest score on negative class for negative examples.
- Latency-aware max-pooling loss discards frames that do not meet latency requirement as shown in the following equation and figure.

$$\mathcal{L}(t_l) = -log(p_{yt}),\;\; t = \begin{cases} argmin_i\,(p_{0i}) & y = 0 \\ argmax_{i\in(i-e<=latency)}\,(p_{yi}) & y \neq 0\text{'} \end{cases}$$

p_{yi} : the probability of class y on frame i.

e : the end of the target word.

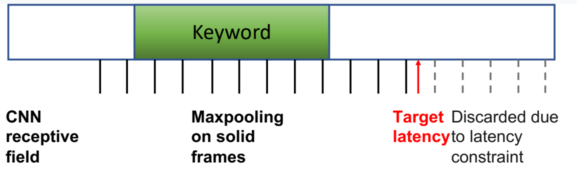
Keyword
CNN receptive field
Maxpooling on solid frames
Target latency
Discarded due to latency constraint

Fig 2: illustration of latency-aware max-pooling loss on positive example

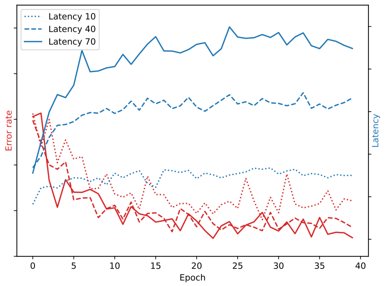

Fig 3: Compare error rate and latency over different training epochs for three models with target latencies, 10, 40, 70. Over training epochs, models learn to trade latency for accuracy. The proposed latency-aware max-pooling loss can control latency accuracy trade-off effectively.

Experimental Results

- Speculation models from both the single task baseline and USDV have the earliest detection.
- Verification mod- els achieves the lowest FAR with more right context.
- USDV model is able to achieve three tasks with different accuracy and latency trade-off, which validates the effectiveness of the MTL training and latency-aware max-pooling loss.
- USDV model achieves same level of performance as baseline models, which shows that the CRNN architecture has enough capacity to perform all three task simultaneously

Model	FAR (%) @ contant FRR	Latency (s)
Speculation	1.75a	b-0.15
Detection	a	b
Verification	0.80a	b+0.3
USDV-speculation	1.75a	b-0.14
USDV-detection	1.03a	b-0.01
USDV-verification	0.82a	B+0.27

Table 1: Performance comparison between USDV and single task baselines.

Conclusions

- We propose an CRNN-based unified speculation, detection, and verification keyword detection model.
- We propose a latency- aware max-pooling loss, and show empirically that it teaches a model to maximize accuracy under the latency constraint.
- A USDV model can be trained in a MTL fashion and achieves different accuracy and latency trade-off across these three tasks.