NTU System at MediaEval 2015: Zero Resource Query by Example Spoken Term Detection Using Deep and Recurrent Neural Networks

Cheng-Tao Chung
Graduate Institute of Electrical Engineering, National Taiwan University
b97901182@gmail.com

Yang-de Chen
Graduate Institute of Communication Engineering, National Taiwan University
yongde0108@gmail.com

ABSTRACT

This note serves as a documentation describing the methods the authors of this paper implemented for the Query by Example Search on Speech Task (QUESST) as a part of MediaEval 2015. In this work, we combined DTW, DNN and RNN in one framework to perform query by example spoken term detection in a zero resource setting.

1. INTRODUCTION

Participants of the task were asked to implement a query by example spoken term detection system on a corpus provided by the organizers. The queries were divided into the development set and the evaluation set, and the list of correct documents are given for the development queries. A soft score and a hard decision for every query-document pair in the evaluation set has to be provided. Note that in this task, only whether or not the query appears in the document is considered, when the query appears in the document is not important. For more information please refer to the overview paper [1].

In this work, we approached the task under a zero resource setting [2] using neural networks. This means we did not use any other information than the corpus itself. For the task considered here, we need to formulate an objective that compares two feature sequences of a query and a document both of varying length and return a score. The Deep Neural Network [3] (DNN) is a state-of-the-art architecture that has been widely applied in speech recognition. However, it is limited to framewise objectives where the length of the input feature has to be fixed. Hence we need to focus on two issues in the work: dealing with the varying sequence length and the formulation of a sequence objective.

Dynamic Time Warping [4] (DTW) is one of the earliest techniques applied in the field and can find the alignment of two sequences, hence transforming both sequences into a feature representation of the same length. DTW solves the problem of varying sequence length. On the other hand, we use Recurrent Neural Networks [5] (RNN) to generate a sequence objective for the query-document pair. In this work, we combine DTW, DNN and RNN in one framework.

2. OBJECTIVE AND APPROACH

Let the feature sequence of an utterance be denoted as $X \in \mathbb{R}^{S \times F}$, where S denotes the length of the sequence, and F denotes the number of dimensions of the feature. We extract 39 dimensional MFCCs with energy, delta and double deltas with HTK[6] for our features in this work. By performing sub-sequence Dynamic Time Warping on the two feature sequences of a document X_d and that of a query X_q, we can find the aligned warping sequences $W_d \in \mathbb{R}^{T \times F}$ in the document and $W_q \in \mathbb{R}^{T \times F}$ in the query, where T denotes the length of the warping sequence.

$$W_d, W_q = DTW(X_d, X_q). \tag{1}$$

We forward the feature frames of both X_d and X_q through the same deep neural network. The number of neurons in the DNN is $39, 100, 100, 39$ on each layer from input to output, We use tanh as the activation function of the network.

$$H_d = DNN(W_d), \tag{2}$$
$$H_q = DNN(W_q). \tag{3}$$

On the final layer of the DNN, the output features are then concatenated, then forwarded to a recurrent neural network. The number of neurons in the hidden layer of the RNN is 50, and the sigmoid function is used as the activation function. The output of the RNN at the time frame t is a single score $s_t(q,d)$, $s(q,d)$ is the average of the score along the entire sequence, and T is the length of the sequence:

$$RNN([H_d, H_q]) = [s_0(d,q), s_1(d,q), ..., s_{T-1}(d,q)] \tag{4}$$
$$s(d,q) = \frac{1}{T}\sum_t s_t(d,q). \tag{5}$$

For every query q, the score of a positive document d_p containing q should be high; the score of a negative document d_n containing q should be low. Therefore, the following is the objective which we wish to minimize:

$$L_q = \sum_{p,n} s(d_n,q) - s(d_p,q). \tag{6}$$

The final objective we wish to minimize is the sum of the objective of all the queries:

$$L = \sum_q \sum_{p,n} s(d_n,q) - s(d_p,q). \tag{7}$$

We train the entire network including both the DNN and RNN using back-propagation algorithm. We take $s(d,q)$ as the score for the document-query pair (d,q), and query would be considered to be in a document if $s(d,q) > 0.5$.

MediaEval 2015 Workshop Sept. 14-15, 2015, Wurzen, Germany

Table 1: Cnxe Results

set	method	Actual Cnxe			
		ALL	T1	T2	T3
dev	dtw	2.0066	2.0064	2.0077	2.0055
	rnn	2.0066	2.0064	2.0077	2.0055
eval	dtw	2.0067	2.0070	2.0093	2.0029
	rnn	2.0067	2.0070	2.0093	2.0029

3. EXPERIMENTS AND RESULTS

The entire corpus was trained only using the corpus of QUESST 2015. We derived two sets of scores from the method above. The first set of scores is the DTW scores generated when we initially align the features between query and document. The second set of scores is the score generated from our RNN in equation 4. We did not perform any pretraining on the network and used random initialization for all the weights. The neural network was implemented using the Theano library [7]. Positive examples for query-document pairs were selected from all query types(T1, T2, T3) in the development set, negative examples were randomly generated query-document pairs. The results of our experiments are shown in Table 1. We only show the actual normalized cross entropy (Cnxe). From the results, we see that the RNN did not perform better than the DTW, and neither system seemed to have performed well. This could be due to error in the implementation or insufficient number of epochs during the training for the RNN. Since the results of the RNN were based on the results of the DTW, it is unclear of what caused the problem.

4. CONCLUSION

The authors of the paper attempted a framework to combine DNN, RNN and DTW under a single zero resource neural network framework for query by example spoken term detection.

5. SUPPLEMENTARY MATERIAL

Since we were encouraged to discuss other systems that we've tried, we've included several versions of our system through different development iterations. All of these systems except for the correspondence auto-encoder have been implemented, yet not all have been evaluated on the corpus. The philosophy behind all these designs was to map query-document pairs to a single trainable objective so the error back-propagates through the entire network.

In the first attempt, we learned feature transformation on the acoustic features (MFCC) using a DNN. The objective of the DNN was a the warping distance after DTW, and the error of the DTW was back-propagated into the network. This design was abandoned due to unreasonably slow calculation: DNN is GPU intensive while DTW is CPU intensive, making this hybrid system hard to implement using Theano.

In the second attempt, we replaced DTW with Convolutional Neural Networks (CNN) [8]. CNNs are GPU friendly architectures which fixes the problem of the previous hybrid systems. The 2D feature representation of every utterance was treated as an image. The acoustic features from different queries/documents were padded with zero vectors to be the same length. This feature transformation CNN only has convolutional layers. The end result the first CNN was

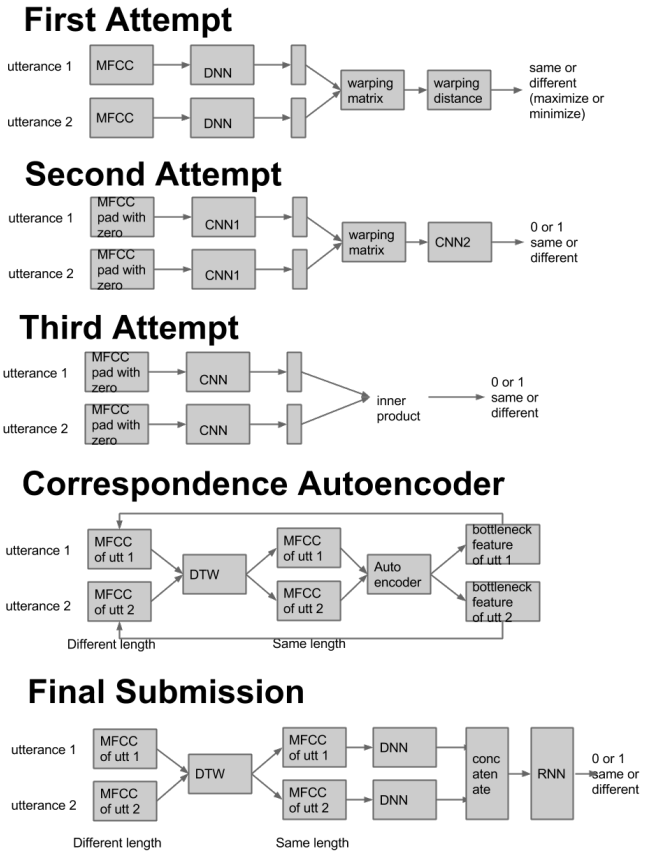

Figure 1: Other systems that we've tried.

another 2D feature representation with one axis being time. We treated the features at the output as if they were acoustic features and plot the warping matrix (pairwise consine similarity). However, instead of applying DTW, we treated the warping matrix itself as another image and forward it through another CNN. The target of the CNN was whether or not the document contains the query. This design didn't work because the error on the testing set didn't converge, maybe due to serious over-fitting.

In the third attempt, we removed the second CNN to reduce the number of parameters. The first CNN has a fully connected layer in this design, and we took the inner product of the fully connected layer from the document and the query to be the error. Although the number of parameters have been reduced, the error on the testing set still didn't converge. Maybe the number of correct training pairs just wasn't enough to train such systems.

Finally, we decided to consider using correspondence autoencoders [9] for this task. The plan was to use correspondence autoencoders as an zero-resource feature extractor. We planned to perform another DTW on these extracted features. Although the system contains both DTWs and DNNs, they are not jointly trained so the performance problem of the first design doesn't occur. However, DTWs are extremely time consuming operations opposed to RNNs. We ran out of time to perform the second DTW so decided to replace it with a RNN which is the system that we submitted in the end.

6. REFERENCES

[1] Igor Szoke, Luis J. Rodriguez-Fuentes, Andi Buzo, Xavier Anguera, Florian Metze, Jorge Proenca, Martin Lojka, and Xiao Xiong. Query by example search on speech at mediaeval 2015. In *Working Notes Proceedings of the Mediaeval 2015 Workshop, Wurzen, Germany*, 2015.

[2] Aren Jansen, Emmanuel Dupoux, Sharon Goldwater, Mark Johnson, Sanjeev Khudanpur, Kenneth Church, Naomi Feldman, Hynek Hermansky, Florian Metze, Richard Rose, et al. A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition. 2013.

[3] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE, 2013.

[4] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[5] Tony Robinson, Mike Hochberg, and Steve Renals. The use of recurrent neural networks in continuous speech recognition. In *Automatic speech and speaker recognition*, pages 233–258. Springer, 1996.

[6] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. *The HTK book*, volume 2. Entropic Cambridge Research Laboratory Cambridge, 1997.

[7] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX, 2010.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] Herman Kamper, Micha Elsner, Aren Jansen, and Sharon Goldwater. Unsupervised neural network based feature extraction using weak top-down constraints.