BUT2012 APPROACHES FOR SPOKEN WEB SEARCH - MEDIAEVAL 2012

Igor Szöke
Speech@FIT, Brno University of Technology, Czech Republic
szoke@fit.vutbr.cz

Michal Fapšo
Speech@FIT, Brno University of Technology, Czech Republic
ifapso@fit.vutbr.cz

Karel Veselý
Speech@FIT, Brno University of Technology, Czech Republic
iveselyk@fit.vutbr.cz

ABSTRACT

We submitted two approaches as the required runs: Acoustic Keyword Spotting as the primary one (AKWS) and Dynamic Time Wrapping as the secondary one (DTW) for the Spoken Web Search task. We aimed at building a simple phone based language-dependent system. We experimented with universal context bottle-neck neural network classifier with 3-state phone posterior features or bottle-neck features.

Categories and Subject Descriptors

H.3.3 [**Information systems**]: Information Storage and Retrieval, Information Search and Retrieval, Search process

General Terms

Experimentation

Keywords

query-by-example spoken term detection, acoustic keyword spotting, dynamic time warping, spoken web search

1. MOTIVATION

Our motivation was to build quickly a simple language-dependent query-by-example spoken term detection system. We counted only on provided development data [1] (audio and phoneme based force alignment). We used only the phone level force alignment from provided development data, so other resources were used.

2. FEATURE EXTRACTION

Firstly, we merged several phonemes. The original alignment consists of 75 unique phonemes. According to amount of available training data for each phoneme (at least 10 seconds) and phone set documentation[1], we reduced the phone set to 50 phones. Then, we trained a simple monophone HMM model and generated phoneme state alignment of the data. This state alignment was used for training of the neural net based classifier. We did not use any other information (spoken language, graphemic transcript, dictionary, etc.).

[1]http://hlt.mirror.ac.za/Phoneset/Lwazi.Phoneset.1.2.pdf

MediaEval 2012 Workshop, October 4-5, 2012, Pisa, Italy

Our feature extractor outputs 3-state phone posteriors or bottle-neck features encoding speech (queries and utterances) in low dimensional feature vectors. The feature extractor is the same as that presented in [2] and contains a Neural Network (NN) classifier with a hierarchical structure called *bottle-neck universal context network*. Energies from 15 Mel-scale critical bands, ranging from 64 Hz to 3800 Hz, are extracted and passed through a logarithm. Next, sentence mean normalization is performed. We obtained a log-critical band spectrogram (CRB), from which long temporal patterns of length 15 are extracted. Hamming window and dimensionality reduction by DCT to six coefficients are applied to each long temporal critical band trajectory.

The input of the context network is a context of 15 frames around the current one, each represented by six DCT coefficients. The input size is $15 \times 6 = 90$. The context NN is a so-called *bottleneck network*. It is trained as a five-layer network with the 3^{rd} layer as the bottleneck of size 80 neurons. The size of the 2^{nd} and 4^{th} layers are 1282 and the number of outputs (5th layer) is 150. It corresponds to the number of phone states - each phone has three phone states.

After training the context network as a 5-layer network, the fourth and fifth layers are cut-off so the output size of the context network is 80.

The merger receives five context net outputs sampled every five frames (for frame t, this is $t-10, t-5, t, t+5, t+10$), so it actually "sees" a 310ms context in the CRB matrix, and the merger input size is $5 \times 80 = 400$. The merger is one of the following:

- A 5-layer phone posteriors bottle-neck NN. It's output consist of the 150 phone states. The size of the bottleneck is fixed to 30 (the 3^{rd} layer) and the size of the 2^{nd} and 4^{th} layers are 1282. This merger produces three state phone posterior probabilities as the output.
- A 3-layer bottle-neck NN. It's output consist of 30 neurons bottleneck (the 3^{rd} layer) and the size of the 2^{nd} layer is 1282. The network is derived from the previous one by cutting-off the 4^{th} and 5^{th} layers. This merger produces 30 bottle-neck features as the output.

Because the lack of speaker information in the development data, we used jackknifing (splitting data into 6 "independent" sets) for devQ-devC condition. We trained on all development data for evaluation condition.

We built a phone recognizer on the above mentioned 5-layer phone posteriors bottle-neck NN. We did not used any phoneme language model - only free phone loop. We achieved phone accuracy 66.02% on the development data (using jackknifing).

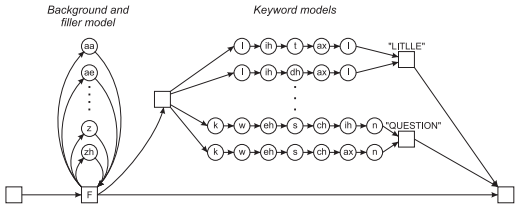

Figure 1: Keyword spotting network used for R-AWKS and AWKS systems.

3. APPROACHES

3.1 Reference Acoustic Keyword Spotting (R-AKWS)

This approach was our reference "upper-bound" system to evaluate the accuracy we can approximately achieve. The R-AKWS follows our paper [4], where we built HMM for each query and then calculated log likelihood ratio between a query model and a background model (free phone loop) - see figure 1. We used the development force alignment and graphemic transcription of queries to obtain reference pronunciation of each query. We achieved Max. TWV 0.737 and the Upper-bound Term Weighted Value (UBTWV) [3] 0.859. The UBTWV finds the best threshold for each query (maximizes the TWV per query) and then averages the scores. It can be considered as non-pooled Max. TWV and shows the room for calibration improvement.

3.2 Acoustic Keyword Spotting (AKWS)

The Acoustic Keyword Spotting is similar to the reference one (R-AKWS). Only we did not use any prior knowledge of queries (pronunciation). The pronunciation of the queries was automatically generated using the above mentioned phone recognizer. After generating pronunciations (one per query), we just striped out surrounding `sil` phonemes. We got Max. TWV 0.453 and UBTWV 0.600 without any score calibration on the devQ-devC condition. Using calibration, we achieved the Max. TWV 0.493 on the devQ-devC condition.

3.3 Dynamic Time Warping (DTW)

DTW system was based on a simple template-matching algorithm on bottle-neck features, where cosine distance was used as a similarity function [5].

4. FILTERING AND CALIBRATION

The primary metric used in these evaluations, the Actual Term Weighted Value (ATWV), is quite sensitive to a good calibration of scores of detections. For each query, we computed an ideal hard-decision threshold and we also generated a feature vector representing the query. Then we trained a linear regression model for predicting the ideal hard-decision threshold. The query feature vector includes:

Length - total length of the query in frames excluding silence around it.

Length of sil - total length of all sil phonemes in frames inside the query.

Phonemes count - Number of phonemes in the query.

Detections count - Number of all detections of the query in devC or evalC data.

Score average global - summed likelihood of all phonemes in the query, divided by the number of total frames.

Score average by phonemes - likelihood of each phone is divided by the number of its frames. These per-frame likelihoods for each phoneme are summed up and divided by the number of phonemes.

We also augmented the feature vector with a square of each feature, which brought a slight improvement.

For training, we used a jackknifing method. We split the development queries into 10 parts and used 9 of them for training and 1% for cross-validation. This was done 10 times for all combinations of the splits. Then we averaged weights of those 10 trained linear models.

Our linear models were trained in Matlab using standard functions `regress` or `robustfit`.

5. RESULTS AND DISCUSSION

Results for the required runs [1] are given in Table 1. We clearly see that the AKWS approach outperforms DTW. We achieved approximately 13% of Max. TWV relative improvement with score calibration on evalQ-evalC condition. The upperbound improvement could be approximately 41% - UBTWV 0.665. We lost approximately 40% relative by switching from hand made pronunciations (R-AKWS) to automatically derived pronunciations (AKWS). The AKWS significantly outperforms DTW also in terms of speed.

Approach	devQ-devC	evalQ-devC	devQ-evalC	evalQ-evalC
R-AKWS	0.786(0.739)	-	0.703(0.653)	-
AKWS	0.493(0.452)	0.629(0.608)	0.429(0.377)	0.530(0.470)
DTW	0.468(0.400)	0.481(0.458)	0.383(0.316)	0.488(0.426)

Table 1: *Max. TWV results for the approaches. Results of non-calibrated system are in brackets. Dx-Qy denotes the set of x data searched for the set of y queries.*

6. ACKNOWLEDGMENTS

Igor Szöke was supported by Grant Agency of Czech Republic post-doctoral project No. GP202/12/P567. This work was partly supported by Czech Ministry of Trade and Commerce project No. FR-TI1/034, Technology Agency of the Czech Republic project No. TA01011328, Czech Ministry of Education project No. MSM0021630528 and IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

7. REFERENCES

[1] F. M. et al. The spoken web search task. In *MediaEval 2012 Workshop*, Pisa, Italy, October 4-5 2012.

[2] F. Grézl and M. Karafiát. Hierarchical neural net architectures for feature extraction in asr. In *Proceedings of INTERSPEECH 2010*, volume 2010, pages 1201–1204. International Speech Communication Association, 2010.

[3] I. Szöke. *Hybrid word-subword spoken term detection*. PhD thesis, 2010.

[4] I. Szöke, P. Schwarz, L. Burget, M. Karafiát, P. Matějka, and J. Černocký. Phoneme based acoustics keyword spotting in informal continuous speech. *Lecture Notes in Computer Science*, 2005(3658):8, 2005.

[5] J. Tejedor, I. Szöke, and M. Fapšo. Novel methods for query selection and query combination in query-by-example spoken term detection. In *Proceedings of Searching Spontaneous Conversational Speech*, pages 15–20, Florence, Italy, 2010.