The JHU-HLTCOE Spoken Web Search System for MediaEval 2012

Aren Jansen, Benjamin Van Durme, Pascal Clark
Johns Hopkins University Human Language Technology Center of Excellence
810 Wyman Park Drive, Baltimore, MD 21211 USA
aren@jhu.edu, vandurme@cs.jhu.edu, pclark@jhu.edu

ABSTRACT

We present an overview for a truly zero resource query-by-example search system designed for the 2012 MediaEval Spoken Web Search task. Our system is based on the recently proposed randomized acoustic indexing and logarithmic-time search (RAILS) framework. The input is merely the raw acoustic observations for the query and search collection, requiring no trained models whatsoever, not even unsupervised ones. Even still the system is capable of search speeds of at least a thousand times faster than real time, and capable of producing competent zero resource performance.

1. INTRODUCTION

Traditional high resource search systems rely on preprocessing the search collection audio with phonetic or word recognizers and constructing efficient indices of these symbolic units to allow fast and accurate retrieval. Our entry into the 2012 MediaEval Spoken Web Search task (see [1] for a complete description of the task and data), is a zero resource approach (no language- or application-specific training resources) that replaces an index of language-specific symbolic units with an index of the individual acoustic feature vectors themselves. The technique, dubbed Randomized Acoustic Indexing and Logarithmic-time Search (RAILS) [3], provides a means to achieve sublinear search times in the absence of a speaker independent symbolic representation, and thus avoids the need for either supervised or unsupervised acoustic models. This permits zero resource search with no upfront training costs and means that RAILS enables, for the first time, keyword search that is both scalable and completely language independent.

2. SYSTEM OVERVIEW

We begin with a brief description of the RAILS approach, a scalable extension of the original segmental dynamic time warping (DTW) approach [5], followed by a description of additional system processing steps introduced specifically for our 2012 MediaEval system. Note that a complete RAILS specification can be found in [3].

2.1 RAILS Overview

The RAILS approach involves four primary processing stages: (1) we map each frame to a sortable bit signature using locality sensitive hashing (LSH) [2], using the variant that preserves cosine distance; (2) we construct sorted lists (the index) of the signatures in the search collection; (3) using the index, we compute approximate nearest neighbor sets for each *query* frame in logarithmic time, allowing the construction of a sparse similarity matrix between query and search collection; and (4) we search for runs of similar frames with efficient sparse image processing techniques applied to the similarity matrix. In our MediaEval system development we investigated system performance as a function of only two RAILS parameters: the size of the neighborhood search beam, B, and the cosine similarity threshold for a frame-level comparison to make it into the sparse matrix, δ. The default values listed in [3] were used for all other RAILS parameters.

2.2 Acoustic Front-End

Based on success for the task demonstrated in [3], we opted to use short-time frequency domain linear prediction features (FDLP-S) [4] for our MediaEval 2012 system. However, we explored the effect of various FDLP-S parameter values on the search task. Inspired by an as-yet-unpublished finding of Vijayiditya Peddinti and Hynek Hermansky during the 2012 JHU CLSP Summer Workshop, we found that decreasing both LPC model order and number of cepstral coefficients produced substantially better speaker independence. We believe this is a product of the resulting spectral smoothing, which reduces speaker dependent effects without substantially reducing phonetic discriminability.

2.3 Query Preprocessing

The nature of this year's query sets are such that some preprocessing was necessary to achieve performance comparable to that previously documented for the RAILS approach. In our previous study [3], all queries were single words and all query examples were extracted from continuous speech using forced alignments. In the present case, queries can contain multiple words (with silence often occurring between and around the constituent words) and are provided in citation form. Both of these factors necessitate the use of speech activity detection to (1) remove begin/end silences from each example and (2) split multiword queries into subsegments about substantial pauses greater than 200 ms. Splitting is thus limited to multiword queries, but note that not all multiword queries required a split. We used a prototype speech activity system being developed at JHU-HLTCOE to identify non-speech regions, though manual speech activity corrections on the query examples were performed as necessary (subsequent improvements in our speech activ-

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

ity system will remove the need for manual intervention). When multiple segments were identified for a given query, each was processed by the RAILS system independently as if they were separate queries. However, the detections for multiword queries only were triggered with the constituent subsegments were detected in the proper order with at most 0.5 s of silence in between.

2.4 Score Normalization and Combination

The requirements of the spoken web search task are such that hit lists for queries are merged before scoring. Therefore, it is essential that scores are properly normalized across query types. Moreover, given we split individual queries into multiple segments using speech activity preprocessing, we require a means to normalize across single-segment and multi-segment queries. During development, it became exceedingly clear that reliable score normalization was essential for good system performance, especially for high precision operating points (e.g. the devA scoring condition).

First, we observed the DTW score distribution for the false alarms is Gaussian. Second, for a sufficiently low threshold, the score distributions were massively dominated by false alarms. Therefore, we can convert each raw DTW score to a normalized z-score using the overall mean and standard deviation estimated from scores of all putative hits for a given query. We define the raw score for multi-segment queries as the sum of the raw DTW scores for each individual segment. However, to make these scores comparable to the single-segment queries, we must again perform a z-normalization as follows. Let q be the number of the segments in a given multi-segment query. First we compute the z-scores for each of the query segments as described above, which we denote $\{z_1, \ldots, z_q\}$. Then, since the sum of two Gaussian distributed random variables is also Gaussian, we can easily compute the z-score for the multi-segment query according to

$$z_{\text{sum}} = \frac{\sum_{i=1}^{q} \sigma_i z_i}{\sqrt{\sum_{i=1}^{q} \sigma_i^2}},$$

where each σ_i is the standard deviation of raw DTW scores for the ith segment of the query.

3. EXPERIMENTS

Facilitated by the efficient search speeds and limited search collection, we conducted extensive experimentation using the development queries on the development search collection. The primary area of interest was the investigation of the RAILS parameters on system accuracy and search speed. Ultimately, we found performance to largely saturate at B=1000 and δ=0.0 (dev/dev MTWV of 0.381), where a real-time speedup of approximately 1000X was observed (this was the system parameters used for the official submission, measured using a 0.8 s long query example). However, there were still marginal gains to be had for a larger $B = 2000$ (0.404 MTWV), but that comes at about half the search speed. Note that increasing δ to 0.25 with $B = 1000$ led to a more dramatic performance drop to 0.331 MTWV, without as substantial improvement in speed (1600X). Note that the real-time speedups are somewhat smaller than those reported in [3], a result of longer queries and smaller search collections used here (recall search speeds are logarithmic in the size of the search collection).

Table 1: Official performance results for all pairings of query set and search collection.

Query Set	Collection	MTWV	ATWV
dev	dev	0.381	0.381
dev	eval	0.336	0.321
eval	dev	0.439	0.421
eval	eval	0.384	0.369

The official results on all four combinations of query sets and search collections are provided in Table 1. Both the actual and maximum term weighted value (ATWV and MTWV) are shown. Note that the score threshold was chosen to maximize TWV using the development query set and search collection, explaining the equivalence of ATWV and MTWV for that case. However, we find that there is a relatively small difference (less than 0.02) between ATWV and MTWV for all conditions, indicating that our score normalization procedure provides a reliable confidence measure for DTW-based search systems.

Ultimately, for truly zero resource systems, a recall ceiling is imposed according to the degree of speaker and channel invariance afforded by the acoustic front end. While progress is being made, the simple truth is that without annotated data in the language of interest (or a closely related language), high recall focused metrics like ATWV will always lag behind. However, zero resource solutions such as the one described here remain lighter weight and more language independent, which make them ideal for many downstream applications.

4. CONCLUSIONS

We have presented an overview of the JHU-HLTCOE system for the MediaEval 2012 Spoken Web Search task along with the evaluation results. The recall of our system is not as high as other zero resource approaches that rely either on borrowing highly supervised acoustic models from other languages or training unsupervised acoustic models for the search collection. However, our system is arguably the most versatile, not requiring the prerequisite of any in-language, out-of-language, or in-domain resources of any kind (not even unannotated data). Furthermore, the efficiency of our indexing procedure permits unprecedented scalability for DTW-based approaches.

5. REFERENCES

[1] F. Metze et al. The spoken web search task. In *Proc. MediaEval*, 2012.

[2] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

[3] A. Jansen and B. V. Durme. Indexing raw acoustic features for scalable zero resource search. In *Proc. Interspeech*, 2012.

[4] S. Thomas, S. Ganapathy, and H. Hermansky. Recognition of reverberant speech using frequency domain linear prediction. *IEEE Signal Processing Letters*, pages 681–684, 2008.

[5] Y. Zhang and J. R. Glass. Towards multi-speaker unsupervised speech pattern discovery. In *Proc. of ICASSP*, 2010.