The SPL-IT Query by Example Search on Speech system for MediaEval 2014

Jorge Proença, Arlindo Veiga, Fernando Perdigão
Instituto de Telecomunicações, Coimbra, Portugal
Electrical and Computer Eng. Department, University of Coimbra, Portugal
{jproenca, aveiga, fp}@co.it.pt

ABSTRACT

This document briefly describes the system submitted by the Speech Processing Lab of Instituto de Telecomunicações, pole of Coimbra (SPL-IT) to the Query by Example Search on Speech Task (QUESST) of MediaEval 2014. Our approach is based on merging results of a phoneme recognition system using three different languages. A version of Dynamic Time Warping (DTW) using posteriorgram distances was created to allow finding some of the peculiar search cases of this task. Our primary submission merges two approaches: simple DTW for detecting entire queries and a version where cutting final portions of queries is allowed. The late submission merges 5 approaches that account for all the search possibilities described for the task, though improved results were only observed in the evaluation dataset for type 3 queries.

1. INTRODUCTION

This year's MediaEval challenge for audio query search on audio, QUESST [1], brings some novelties to the table, and further details can be consulted in the referenced paper. Shortly, the queries do not have to match exactly in the searched audio. They can present small changes, extra words in between and a reordering of the words. For such, we devised different strategies to target these cases. Our late submission targets all of these cases, whereas our first and primary submission only targets exact and different-end matches, while still operating relatively well.

Our motivation is to continue to explore and research methodologies related to word-spotting, and we foresee that what is learned from this task is certainly going to be useful for future projects.

2. SYSTEM DESCRIPTION

2.1 Phonetic recognizer

We started by using our in-house phoneme recognizer for Portuguese, which is based in Hidden Markov Models, with our keyword spotting system [2]. Since it was hard to obtain posteriorgrams with this technique, we decided to use an available external system based on neural networks, the phoneme recognizer from Brnu University of Technology (BUT) [3]. We used the three available systems for 8kHz audio, for three languages: Czech, Hungarian and Russian. Using different languages allows us to deal with different sets of phonemes, and hopefully the fusion of the results will better describe the similarities between what is said in a query and in the searched audio.

All queries and audio files were run through the 3 systems, resulting in phoneme state posteriograms (3 states per phoneme). Leading and trailing silence/noise were cut on queries, from the initial and final frames that had a high probability of corresponding to silence or noise (sum of the 3 states of 'int', 'pau' and 'spk' phones is greater than 50% for the average of the 3 languages).

2.2 Dynamic Time Warping

We implemented a version of Dynamic Time Warping (DTW) specific for this challenge. As in [4], the local distance is based on the dot product of query and audio posterior probability vectors. Also, a back-off of phoneme probabilities with lambda=10^{-4} is applied, and minus log is applied to the dot product. This results in the local distance matrix for DTW.

The beginning and end of a DTW path was not restricted. For the local path restrictions we tested a small number of alternative options, but the most versatile was found to be the one that allows a path to continue through 3 jumps to directly adjacent points in the matrix: horizontal, vertical and diagonal. All these 3 types of movements have equal (unitary) weight. The final path distance is normalized by the number of query frames. This simple approach (named A1) will output the distance of the best path and is the basis from which the subsequent approaches will be devised.

2.3 Modifications on the DTW

To account for the special types of queries indicated in the task, we developed 4 additional approaches based on altering the DTW:

(A2) This approach considers a cut of up to 250ms at the end of a query, keeping the non-cut segment above 500ms (example on Figure 1).

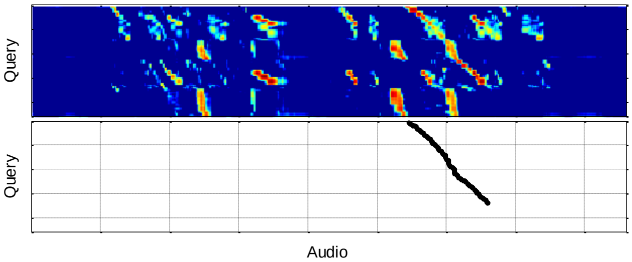

Figure 1. Query vs. Audio posterior distance matrix (top) and the best path from A2 (bottom).

(A3) This approach considers a cut at the beginning of a query up to 250ms, keeping it at least above 500ms. To improve computational speed, we reason that the basic DTW paths that include the matching query should already have one of the lowest distances for this query-audio pair, therefore we only search solutions by backtracking the 5 best total paths.

(A4) This approach allows just one "jump" in the DTW path (Figure 2) - for each possible path, a jump of up to half the query's length is allowed to cover for possible extra words between the query's own words. The jump may not occur at the initial and final 250 ms of the query, and for queries shorter than 800ms.

MediaEval 2014 Workshop, October 16-17, 2014, Barcelona, Spain

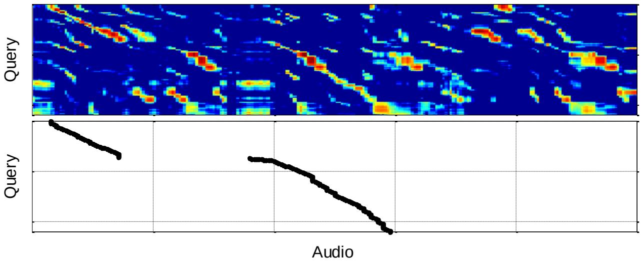

Figure 2. Query vs. Audio posterior distance matrix (top) and the best path from A4 (bottom).

(A5) This approach allows swaps of two path segments (Figure 3). This accounts for re-ordering of query words by backtracking the candidate DTW paths from the end of query (as for A3) and finding an alternative path that appears ahead of the initial one, but which better matches the beginning of the query. This second path segment can't start before the end of the first one but can start later to account for a gap due to an extra word in the middle of the query. The same limitations as for A4 apply.

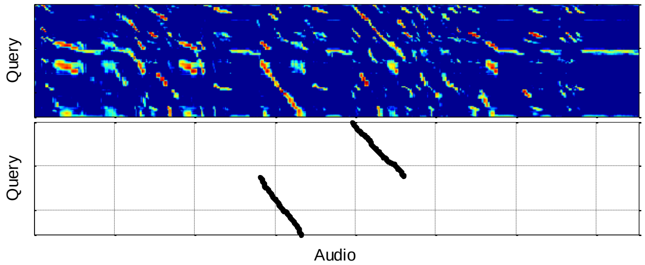

Figure 3. Query vs. Audio posterior distance matrix (top) and the best path from A5 (bottom).

All examples in the figures are true cases from the development dataset that were rejected at first with strategy A1 but were now accepted with one of the other approaches.

2.4 Fusing

Since different approaches provide different distance measures for the same query-audio pair, one could argue that the minimum of distances obtained through them would correspond to the best detection possible. However, tests showed that the minimum was not the best method, supposedly increasing false alarms through one of the special approaches. The harmonic mean was found to be a good compromise, and was employed here to extract a single distance value from several approaches.

Per-query normalization is performed, by subtracting the mean and dividing by the standard deviation of all the results from Query-Audio pairs for a given query. This step may skew the results to indicate that every query should be found at least once on the database, but we found that this procedure was highly beneficial.

To fuse systems that are based on recognizers of different languages, we employ the arithmetic mean of the already normalized values, which was found to be the best method on the development dataset. Distances are transformed into figures of merit simply by taking their symmetrical values.

2.5 Processing Speed

The hardware that processed our systems was the CRAY CX1 Cluster, running windows server 2008 HPC, and using 16 of 56 cores (7 nodes with double Intel Xeon 5520 2.27GHz quad-core and 24GB RAM per node). Approximately, the Indexing Speed Factor was 1.4, Searching Speed Factor was 0.0029 per sec and per language, and Peak Memory was 0.098 GB.

3. SUBMISSIONS AND RESULTS

We submitted two systems for evaluation, one primary on-time and one late. The primary system is simply a fusion of the A1 and the A2 approach for the 3 languages. The late system corresponds to the fusion of the 5 approaches. The summarization of the scores obtained for each of them is shown on table 1.

Table 1. Summarization of the obtained results on development and evaluation datasets.

	primary	late
Cnxe, MinCnxe - Dev	0.6797, 0.5438	0.7106, 0.5881
Cnxe, MinCnxe - Eval	0.6588, 0.5080	0.6708, 0.5240
ATWV, MTWV - Dev	0.4494, 0.4494	0.4051, 0.4052
ATWV, MTWV - Eval	0.4399, 0.4423	0.3918, 0.4218

The late system did not improve overall results for all matching query types. By analyzing the results of individual query types, we found that it did improve slightly for type 3 queries, from 0.8049 Cnxe on primary to 0.7865 Cnxe on late systems. This is where approaches A4 and A5 would be exclusively useful and the late system was submitted to include and discuss these special methods that, at first, were significantly increasing the number of false positives in our trials. For each solo approach, the resulting Cnxe scores on the Eval dataset were: A1: 0.6823, A2: 0.6721, A3: 0.6947, A4: 0.6957 and A5: 0.6999.

By analyzing the output from the scoring tool, it is also observed that the Cnxe could have been severely decreased (0.6797 to 0.5438 on Dev set), showing that we lacked an optimization method for it.

4. Conclusions

The complicated nature of this year's task presents an added difficulty and we have tried a few strategies for dealing with it. We should have definitely applied methods to optimize Cnxe, since we know that resembling the attainable MinCnxe would improve our results by a big margin, and is something to review for future participations. Our main conclusion is that including the possibility of, e.g., re-ordering of words, increases false positives overall for our approach, and as these special cases are a small part of the database, results may worsen.

We would like to acknowledge the BUT group for making available their phoneme recognition system. Jorge Proença is supported by the SFRH/BD/97204/2013 FCT Grant

5. REFERENCES

[1] X. Anguera, L.J. Rodriguez-Fuentes, I. Szöke, A. Buzo and F. Metze, "Query by Example Search on Speech at Mediaeval 2014", in Working Notes Proceedings of the Mediaeval 2014 Workshop, Barcelona, Spain, October 16-17

[2] A. Veiga, C. Lopes, L. Sá, F. Perdigão. Acoustic Similarity Scores for Keyword Spotting. In *PROPOR 2014*, São Carlos, Brazil, October 6-9, 2014.

[3] Phoneme recognizer based on long temporal context, Brno University of Technology, FIT, http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context

[4] T.J. Hazen, W.Shen, C.M. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *ASRU 2009*: 421-426.