Telefonica Research system for the Query-by-example task at Albayzin 2012

Xavier Anguera

Telefonica Research,
Edificio Telefonica-Diagonal 00, 08019, Barcelona, Spain,
xanguera@tid.es

Abstract. In this paper we describe the Telefonica Research submission to the Albayzin 2012 Evaluation. In particular, we participated in the query-by-example task within the search on speech evaluation. Our system is a a zero-resources approach by using a variant of the Dynamic Time Warping algorithm. We also preprocess and post-process the signal to make the features as much speaker independent as possible, to eliminate silence frames usually causing an increase in false alarms and eliminating unnecessary overlapping matching segments. The results are low due to the difficulty of the task and the strictness of the used metric, but are representative of the potential of these techniques for search on speech when no information is available a priori on what language or acoustic conditions we are facing.

Keywords: Query by example, pattern matching, low resources

1 Introduction

The query-by-example subtask proposed within the search on speech task in the Albayzin 2012 evaluation proposes to search for occurrences of a given query within a spoken database where the query is given in acoustic form and where no transcription is available neither of the query nor of the spoken database. This task, together with the spoken term detection task where a textual query needs to be searched over the audio corpus, are gaining importance in the last few years and are being applied to cases where not much data is available a priori to train the systems being used. This is the case of languages with few available resources or when encountering very adverse acoustic conditions for which no pretrained models are available. For this talk two kinds of algorithms are usually applied. On the one hand, researchers are looking again into what was being used before the arrival of Hidden Markov Models and improving them to be adapted to the new challenges ahead. This includes algorithms that search for patterns in the speech, like Dynamic Time Warping (DTW).

One of the first algorithms to improve on the classical DTW algorithm was proposed in [8] where an iterative search was done in an audio signal looking for repeating patterns. Later on, other researchers have proposed alternative solutions [11, 10, 9] that allow systems to be more accurate and/or fast. Recently, in

[7] we have seen that this task does not require any more of expensive equipment to run and is very scalable to search on high volumes of data.

An alternative to DTW-like algorithms are the standard phonetic search approaches. If the language spoken in the recordings is known, does not change over time and we have training data for it, very good results can be obtained (for reference, see any of the literature in the spoken term detection community). Problems with these algorithms often come when the language is unknown or we have a low-resources situation, in which similar languages have to be adapted to be able to represent the data being searched, with clear accuracy problems. On their favor, these systems are born scalable as they can use lots of the speech recognition speedup methods that have been proposed over the years, although their computational requirements usually involve heavy machinery (maybe with parallel processors) to be able to process the data as fast as possible. In 2011, the Mediaeval SWS evaluation [5] proposed a query-by-example task on indian data and in [6] results are given comparing both phonetic-based and pattern-matching based approaches.

The approach we applied for this evaluation is a zero-resources approach by using a variation of the DTW algorithm called Segmental-DTW. Such algorithm is able to search for a given query throughout the reference data by allowing a certain time warping between the query and the reference instances of the queried word. In this paper we describe the system implemented and the results we obtained in the evaluation. In general, given the difficulty of the task, we are happy with the results, although there are some possible areas of improvement as we will highlight in the conclusions and future work section.

2 System description

For the query-by-example evaluation we have applied a zero-resources matching approach based on the well-known Dynamic Time Warping (DTW) algorithm. The whole algorithm (from signal input to results output) is depicted in figure 1. First, acoustic features need to be extracted from the signal. For this we first obtain MFCC-39 coefficients from the signal (13 Cepstra + 13 delta + 13 double delta). As shown in [2] the MFCC features themselves are not sufficiently speaker independent. For this reason we use them to compute posteriorgram features by using a posteriors background model we describe in more detail in Section 2.1 below. Once posteriors are computed we trim out those frames that are classified as silence or very low speech as described in Section 2.2. The resulting features are stored on the database for reference features and sent to the matching algorithm for the query features. Section 3 describes in detail the matching algorithm we used for this evaluation. After the matching algorithm returns all possible matching paths and their scores, we postprocess the results to merge all those matches that overlap with each other, as explained in Section 2.3.

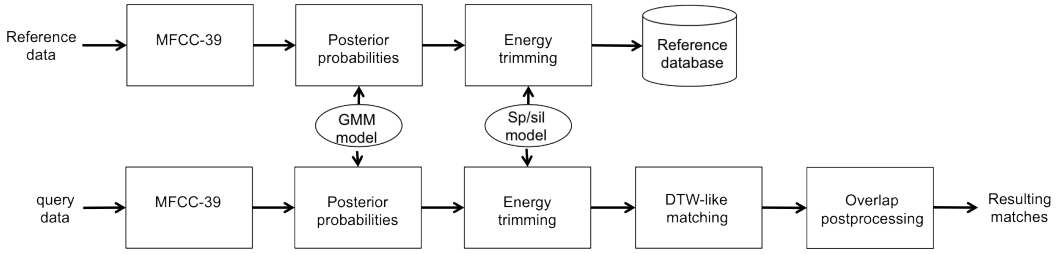

Fig. 1. *General system blocks for QbE task*

2.1 Posteriors Background Model

Posterior probabilities have been successfully used in pattern matching for some time [3]. Several methods have been proposed to obtain the posterior probabilities. In our systems we use Gaussian posteriors obtained from a GMM model that has been previously trained on all available reference data (i.e. development and testing data). The GMM is trained using a combination of EM and K-means iterations in order to maximize the discovery and separation of automatically discovered acoustic regions in the acoustic space. For more information on the model refer to [1].

2.2 Speech/silence labeling

One of the biggest enemies of pattern matching approaches is silence, as silence usually matches very well with silence, thus returning many false alarms unless it can be trimmed back. To eliminate silence from our input data (both queries and references) this year we trained a speech/silence classifier using GMM models in a non-supervised way. First, we gather the 10% of acoustic frames with lowest energy from our training data (the reference files in the development set). With these frames we train a one Gaussian silence model and with the rest we train a 4-Gaussian speech model. Then we iteratively assign each frame in the training set to the closest model and retrain the models. This usually increases the number of frames in the silence model. We stop after 20 iterations or when the difference in number of frames between two consecutive iterations is very small. We store the Gaussians in the speech model ordered by their mean energy. In order to label the data using this model we no only assign each frame to the most likely model, but in the case of speech, we record which of the Gaussian mixtures is the closest one. In test we consider as silence (and therefore do not use for matching) any frame that is labelled as silence or is assigned to the lowest speech Gaussian.

2.3 Matching segments overlap detection

The overlap detection module is used on all matching segments returned by the matching module to reduce the number of segments finally returned as matches.

It is common that matching algorithms based on pattern matching return a high number of possible paths around the true matching path, as frames around the optimum path also contain very low distances. In our implementation we allow the matching algorithm to register all possible paths it finds and then we search for those paths that have an overlap in the reference higher than 50% (note that the query segment is always the same, covering all the query time). In our implementation we payed special attention for matching paths that are contained within bigger paths. These are sometimes specially good subsequences within a longer sequence. Given two paths that are in overlap, in the current system we select that one path that had a higher average score (i.e. in average it matches between the query). In the past we would merge the paths by selecting the smallest and biggest of the start and end points, respectively. We are not following this approach anymore as the matching algorithm requires for some precision in the detection of the matching segment, therefore it is better to end up with shorter paths than with much longer paths.

3 Segmental Dynamic Time Warping

Given two sequences, X and Y of posterior probabilities, respectively obtained from the query and the reference audio recordings, we compare them using a DTW-like algorithm. The standard DTW algorithm returns the optimum alignment between any two sequences by finding the optimum path between their start $(0, 0)$ and end (x_{end}, y_{end}) points. In our case we constraint the query signal to match between start and end, but we allow the phone recording to start its alignment at any position $(0, y)$ and finish its alignment in whenever the dynamic programming algorithm reaches $x = x_{end}$. This algorithm is generally called Segmental Dynamic Time Warping (SDTW). Although we do not set any global constraints in the algorithm (in order to allow any matches to exist in any position in the reference recording), the local constraints used by the dynamic programming are set so that at maximum 2-times or $\frac{1}{2}$-times warping is allowed by choosing the path that minimizes the cost to reach position (i, j) as

$$\text{cost}(i,j) = (d(i,j) + \min \begin{cases} D(i-2,j-1))/(\#(i-2,j-1)+3) \\ D(i-2,j-2))/(\#(i-2,j-2)+4) \\ D(i-1,j-2))/(\#(i-1,j-2)+3) \end{cases} \quad (1)$$

Where $D(i, j)$ is the accumulated (non-normalized) distance of all optimum paths until position (i, j), $d(i, j)$ is the local distance between frames x_i and y_j from both compared sequences, and $\#(i, j)$ is the number of jumps of the optimum path until that point. Note than when normalizing the different possible paths we slightly favor the diagonal match by considering the Manhattan distance between the current point and the previous point. Such inequality is also kept when updating the number of jumps matrix, thus favoring paths that contain many diagonal matches. In the current implementation we run the algorithm independently for each reference file and each query file, but keeping all

of them in memory for fast access to the data. Once the cost matrix has been computed for a given query and reference file, we find in the last row of such matrix (positions (N, j) where N is the last query frame and j are all possible reference frames) the aggregate scores of the best paths matching query and reference data. By locating the minima of this function and applying a threshold we can find the hypothetical matches between our query and the reference. By performing a backtracking of the selected matches we can find where in the reference data the match started. Such position will always be between 2-times and $\frac{1}{2}$-times the length of the query signal, as imposed by the local constraints imposed above.

4 Evaluation Database and Results

As described in [4], data provided by the organizers for system training, development and evaluation belong to recordings of the MAVIR workshops in 2006, 2007 and 2008 (Corpus MAVIR 2006, 2007 and 2008) corresponding to Spanish language. Training/development data amounts about 5 hours of speech material in total. For evaluation, test speech data amounts about 2 hours in total. Queries are acoustic snippets of the data containing words spoken several times in the query and reference files. In general we observed that the query snippets are quite short (less than 1 second) and contain usually just one word per query with no silences at the beginning or end. These queries have been obtained from the reference data by cutting a particular instance of the word, usually uttered within a sentence with other words in coarticulation just before and after it. Although this is not the optimal use case for a query-by-example application, it is a challenging one that systems should be able to handle.

The metrics used are the same as used in the spoken term detection task originally developed by NIST in the STD 2006 evaluation and described also in [4]. There are two main metrics. On the one hand, the Actual Term Weighted Value (ATWV) computes an individual weighted sum of misses and false alarms for each query term and given a working general threshold, and then obtains the average value. On the other hand, the Maximum Term Weighted Value (MTWV) looks at all possible thresholds and selects the optimum one. It is desired that the threshold defined on the development data also be optimal for the evaluation, so that the system can generalize to unseen datasets.

Table 1 shows the official results we obtained in the evaluation for the development and the evaluation sets. The threshold used in the evaluation set is the same one we optimized for the development set. We noticed how our system obtains quite low scores for the MTWV, which is due to the high importance that the scoring gives to the false alarms. On the development set, by the ATWV system, we obtain 45 correct matches, 33 false alarms and 982 misses. It is clear that the false alarms are driving the scoring, which is one of our main current problems.

The relationship between misses and false alarms can be seen more clearly in Figure 2, which shows the DET curve of misses versus false alarms for the

Table 1. Official Evaluation Results

Metric	dev set	eval set
MTWV	0.0455	tbd
ATWV	0.0425	tbd

development set. We can see how our system gives results far better than random, although it still have a long way to go to call this task as solved.

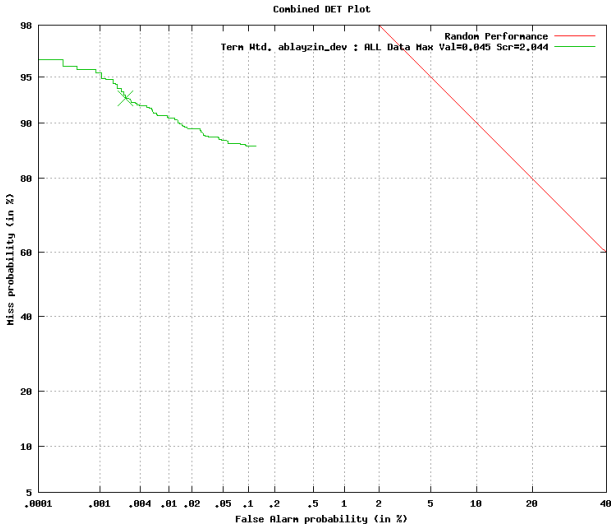

Fig. 2. *DET curve for the development set*

5 Conclusions and Future Work

In this paper we present the Telefonica Research system we presented to the Albayzin 2012 query-by-example task. For this we used a zero-resources approach based on the segmental-DTW algorithm. Our results are far from optimal and indicate how difficult the task is. We are planning on working out some algorithms to reduce the number of false alarms, and therefore should obtain better MTWV in the future.

References

1. X. Anguera: Speaker Independent Discriminant Feature Extraction for Acoustic Pattern-Matching. In Proc. ICASSP 2012, Kyoto, Japan.
2. A. Muscariello, G. Gravier and F. Bimbot: Zero-resource audio-only spoken term detection based on a combination of template matching techniques In Proc. INTERSPEECH 2012, Florence, Italy.

3. Y. Zhang and J. Glass: Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams In Proc. ASRU 2009, Merano, Italy.
4. J. Tejedor, D. T. Toledano and J. Colas: The ALBAYZIN 2012 Search on Speech Evaluation In Proc. Iberspeech 2012, Madrid, Spain.
5. MediaEval 2011 Workshop, Pisa, Italy, Sept. 2011.
6. F. Metze, N. Rajput, X. Anguera, M. Davel, G. Gravier6, C. Heerden, G.V. Mantena, A. Muscariello, K. Prahallad, I. Szoke, and J.Tejedor: The spoken web search task at mediaeval 2011 In Proc ICASSP 2011, Kyoto, Japan.
7. A. Jansen, B. V. Durme: Indexing Raw Acoustic Features for Scalable Zero Resource Search In Proc. Interspeech 2012, Portland, OR, USA
8. A. Park and J. Glass: Unsupervised Pattern Discovery in Speech In IEEE TASLP, vol. 16, no. 1, January 2008
9. A. Jansen, K. Church and H. Hermansky: Towards Spoken Term Discovery at Scale with Zero Resources In Proc. Interspeech 2010, Tokyo, Japan.
10. A. Muscariello, G. Gravier and F. Bimbot: Variability tolerant audio motif discovery In Proc. Int. Conf. on Multimedia Modeling, 2009
11. X. Anguera, R. Macrae and N. Oliver Partial Sequence Matching using an Unbounded Dynamic Time Warping Algorithm In Proc. ICASSP 2010, Dallas, TX, USA