Telefonica System for the Spoken Web Search Task at Mediaeval 2011

Xavier Anguera
Telefonica Research
Torre Telefonica-Diagonal 00
08019 Barcelona, Spain
xanguera@tid.es

ABSTRACT

This working paper describes the system proposed by Telefonica research for the task of spoken voice search within the Mediaeval benchmarking evaluation campaign in 2011. The proposed system is based exclusively on a pattern matching approach, which is able to perform a query-by-example search with no prior knowledge of the acoustics or language being spoken. The system's main contributions are the usage of a novel method to obtain speaker independent acoustic features to later perform the matching through a DTW-like matching algorithm. Obtained results are promising and show, in our opinion, the potential of such class of techniques for this task.

Categories and Subject Descriptors

H.303 [**Information Search and Retrieval**]: Miscellaneous

General Terms

Algorithms, Performance, Experimentation

Keywords

Pattern matching, query-by-example, spoken query, search

1. INTRODUCTION

The objective of the spoken web search task is to search for some given audio query within a set of given audio content, for a detailed explanation refer to [4]. The audio content in this particular evaluation contains phone call excerpts recorded in 4 different languages within the World Wide Telecom Web project [3] conducted by IBM. The system we propose to tackle this task is based on audio pattern-matching between the query and the audio content to retrieve putative matches. No information at all is used regarding the language that the queries are spoken in or the content (i.e. the transcription).

2. SYSTEM DESCRIPTION

The proposed system can be split into two main blocks: the acoustic feature extraction and the query search. For the acoustic feature extraction the goal is to obtain features that contain information about what has been said while they are speaker independent so that the system is able to recognize two instances of the same spoken word, even if they were spoken by different speakers. The query search doe a search for every particular query over all acoustic material to identify whether (and where) the query appears. Transversal to both modules we applied a simple silence detection algorithm to eliminate long silences in the queries and in the audio content. Next we will describe these three modules more in detail.

2.1 Silence Detection and Removal

Early on in our development we noticed that most queries were spoken in isolation. This means that the spoken query is always accompanied with some silence at the beginning and end. In addition, some phone call excerpts also contained non-wanted long amounts of silence frames. In order to eliminate most silence regions without jeopardizing the non-silence ones we applied a simple energy-based thresholding algorithm, individually in every file, as follows: first, we compute the average energy of the signal over windows of 200ms, every 5ms. Then we search for the smallest energy value and the average of the top 1% highest energy values (we do not choose a single value in order to mellow down the effect of outliers) . Next we compute a threshold at the 5% of the resulting dynamic range, above the minimum energy value. Finally we apply such threshold to every 5ms of the input signal to differentiate between speech and silence. To avoid fast changes between speech/silence we apply a top-hat algorithm with a window of 100ms to the binary output of the previous step to ensure that no silence/speech segments are output with less than 100ms length.

2.2 Acoustic Features Extraction

Most of our effort in this year's evaluation went to design a good acoustic feature extraction module. Our goal was to extract from the audio signal some features that retained all acoustic information about what was said while being speaker and background independent. As a side objective, we also wanted to be as much independent as possible to outside training data.

We focused the design of our feature extractor in previous work that started with [1] on using phone posterior probabilities as features, which was then extended by [5] to apply it to the automatic word discovery task. Similarly to [5], for our main submission we construct a Gaussian Mixture Model and store the Gaussian posterior probabilities (normalized to sum 1) as our features. In our case we decided to only use the development data available for the SWS task,

MediaEval 2010 Workshop, September 1-2, 2011, Pisa, Italy

therefore no external data was used on the training of this model. In addition, once the GMM has been trained with the EMML algorithm we perform a hard assignment of each frame to their most likely Gaussian and retrain the Gaussian's mean and variance to optimally model these frames. This last step tries to solve the problem most EMML systems have, which is focusing on optimizing the Gaussians parameters to maximize the overall likelihood of the model on the input data, but not to discriminate between the different sounds in it. By performing the last assignment and retraining step we push Gaussians apart from each other to better model individual groups of frames depending on their location and density.

Alternatively, we also submitted a contrastive system that consists on the binarization of the posterior probabilities for each frame to binary form. This is inspired by our recent developments in speaker verification [2] where we show that we can effectively build binary models to identify between speakers. Such representations are much smaller for storage purposes and can be processed much faster as binary distances are usually very fast. In this case, for every posterior probabilities vector we turned to 1 the 20%-best probabilities, and to 0 the rest. The chosen distance between two binary vectors x and y was defined as

$$S_d(x,y) = \frac{\sum_{i=1}^{N}(x[i] \wedge y[i])}{\sum_{i=1}^{N}(x[i] \vee y[i])} \quad (1)$$

where $\wedge$ indicates the boolean AND operator and $\vee$ indicates the boolean OR operator.

2.3 Query search Algorithm

Given two sequences, X and Y of posterior probabilities, respectively obtained from the query and any given phone recording, 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 $x = x_{end}$. Although we do not set any global constraints, the local constraints 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

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

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.

3. RESULTS AND DISCUSSION

Table 1 shows the official results we obtained with our systems, for the primary (posteriorgrams features) and contrastive (binarized features) submissions. In all cases we report the Term Weighted Maximum Value (TWMV) instead of the actual value as we did not place much emphasis on in the development stage at finding an optimum threshold for our system. Still, we observed that for any given set threshold the results remain similar both in dev-dev and in eval-eval.

Table 1: Term Weighted Max Value for the submitted systems

Dataset-termlist	Posteriorgrams	binary features
dev-dev	0.156	0.205
dev-eval	0.019	0.022
eval-dev	0.000	0.000
eval-eval	0.173	0.222

In general, we find results for dev-dev and eval-eval to be very acceptable. On the other hand we were surprised to see that our system does not work nearly as well for the cross conditions. We have observed that channel missmatch might have played a major role in these results, as we observed in several cases that development files contain many recordings with a much poorer signal quality than those from evaluation files. We consider we have achieved a reasonable speaker independence with our features but we are still to apply ways to compensate for differences in the channel.

Comparing the two submissions we observe that the binary features are always outperforming the standard posteriorgrams. In our point of view this is a very interesting finding that can be used in the near future to speedup the spoken word search and automatic pattern discovery systems, which together with the proposed novel way to compute the GMM model can achieve fast and quite accurate results.

4. REFERENCES

[1] G. Aradilla, J. Vepa, and H. Bourlard. Using posterior-based features in template matching for speech recognition. In *Proc. ICSLP*, 2006.

[2] J.-F. Bonastre, X. Anguera, G. H. Sierra, and P.-M. Bousquet. Speaker modeling using local binary decisions. In *Proc. Interspeech*, 2011.

[3] A. Kumar, N. Rajput, D. Chakraborty, S. K. Agarwal, and A. A. Nanavati. Wwtw: The world wide telecom web. In *Proc. NSDR 2007 (SIGCOMM workshop)*, Kyoto, Japan, August 2007.

[4] N. Rajput and F. Metze. Spoken websearch. In *MediaEval 2011 Workshop*, Pisa, Italy, September 1-2 2011.

[5] Y. Zhang and J. Glass. Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams. In *Proc. ASRU*, pages 398–403, Merano, Italy, December 2009.