An Evolutionary Confidence Measure for Spotting Words in Speech Recognition

Alejandro Echeverría, Javier Tejedor and Dong Wang

Abstract Confidence measures play a very important role in keyword spotting systems. Traditional confidence measures are based on the score computed when the audio is decoded. Classification-based techniques by means of Multi-layer Perceptrons (MLPs) and Support Vector Machines have shown to be powerful ways to improve the final performance in terms of hits and false alarms. In this work we evaluate a keyword spotting system performance by incorporating an evolutionary algorithm as confidence measure and compare its performance with traditional classification techniques based on MLP. We show that this evolutionary algorithm gets better performance than the MLP when False Alarm (FA) is high and always performs better than the confidence measure based on the single score computed during the audio decoding.

1 Introduction

Speech recognition converts spoken words to text. Generally, the automatic speech recognition (ASR) systems try to identify all the words of a target language and produce an output consisting of the words found in the speech signal along with the initial and end times of each. Keyword spotting, as a task within speech recognition, deals with the search of a predefined set of terms within the speech signal. Therefore, it differs from the ASR systems due to just a few terms are important to be identified from the speech signal. Traditionally, keyword spotting systems use a dictionary composed of these important words, modeled from their phone transcriptions, plus

Alejandro Echeverría Rey
Universidad Autónoma de Madrid, Spain, e-mail: alejandro.echeverria@uam.es

Javier Tejedor
Universidad Autónoma de Madrid, Spain e-mail: javier.tejedor@uam.es

Dong Wang
University of Edinburgh, UK e-mail: dwang2@inf.ed.ac.uk

filler models to deal with the non-relevant words of the speech signal [6, 12, 11]. Therefore, their output is composed of the putative occurrences got from the decoding process plus filler models, which are automatically rejected. The main drawback of these systems is the need of re-processing the audio as soon as the list of terms changes, which is the most time-consuming task in a keyword spotting system and impractical when processing hundreds of hours. Alternatively, other keyword spotting systems index the audio in terms of sub-word units (commonly in the way of a lattice, i.e, a graph composed of nodes and arcs that connect the nodes) and next integrate an algorithm to search for the list of terms from such lattice. Although these systems have poorer performance than the filler model-based approaches due to they do not make use of any lexical information, there is no need of re-processing the audio when the list of terms changes, which is, in practice, highly valuable.

In keyword spotting, there are two different classes from which the performance can be measured. An occurrence detected by the keyword spotting system is a "hit" if it appears in the speech signal between the time interval given and a "False Alarm (FA)" when does not. It is straightforward to conclude that a keyword spotting system should produce as many hits as possible while minimizing the number of FAs. Confidence measures based on this idea haven been proposed to improve the system performance. Some of them use the confidence (score) computed during the decoding process of that the occurrence is actually a hit [11, 4], rejecting those whose score falls below a predefined threshold. On the other hand, as the occurrences belong to one of these two classes, hit or FA, confidence measures based on classification techniques have been also studied (e.g. Neural Networks (NNs) and Support Vector Machines (SVMs) [8, 1]) to reject those classified as FA.

On the other hand, evolutionary computation is a global optimization technique broadly applied to diverse fields [5] like schedule optimization, robot navigation, controller design, image processing, discrimination and classification, and so on. Therefore, as the final decision of accepting or rejecting a putative occurrence in keyword spotting is a classification problem, we propose the use of an evolutionary algorithm called Evolutionary Discriminant Analysis (EDA) [10] to classify the occurrences found during the search in lattice with the purpose of rejecting those classified as FA. We compare it with traditional techniques based on NNs (MLP) and on the score computed during the decoding process.

This paper is divided as follows: After this introductory section, the Speech recognition task is introduced in section 2. Section 3 explains how the confidence measures are built. In section 4, the experiments are presented. The last section of the paper presents the conclusions and opens some areas for future work.

2 Speech Recognition

We have focused on the keyword spotting task related to speech recognition. The keyword spotting system consists of three different steps:

Lattice generation: The audio files are decoded in terms of sub-word units (phones in our case) and the phone lattice is stored. Figure 1 shows an example of a phone lattice. The HTK tool [14] has been used for the lattice generation, where the Viterbi algorithm decodes the speech signal and generates the lattice. The lattice is generated by running the Viterbi algorithm in N-best mode and a depth of N=5 was found suitable in preliminary experiments.

Lattice search: An algorithm that searches for the actual phone transcription of each term in the lattice is run in order to generate the putative detections. Along with the detection of each term, and the initial and end times got from the initial and end times of the first and last phone of the term detected, this algorithm also computes a score. It represents the confidence of that each detection is a hit, using the language model score and acoustic score of each phone of the detection. These two values were stored in the lattice during the lattice generation. Readers are referred to [11] for more information about the score computation. The *Lattice2Multigram* implementation provided by the Speech Processing Group, FIT, Brno University of Technology has been used in this step.

Confidence measure: This module classifies the occurrences detected by the previous step in hit or FA and rejects those classified as FA to present the final list of occurrences.

Figure 2 shows the different modules of the whole system.

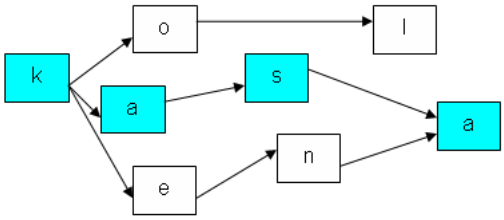

Fig. 1 An example of a lattice where the term "casa", whose phone transcription is "k a s a", has been found by the *lattice search* step.

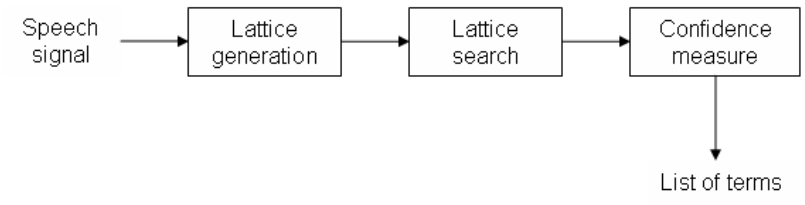

Fig. 2 Modules of the keyword spotting system, which receives a speech signal and outputs a list of terms found in it.

3 Confidence measure

3.1 Feature selection

In building each confidence measure, three term-dependent features have been chosen as an input super-vector to the classification algorithm. It has been proved their good performance in hit/FA classification [13]. These features are:

Score: It represents the confidence of each detection computed as in [11].

Effective occurrence rate: It is defined as illustrated in Equation 1.

$$R_0(K) = \frac{\sum_i c_f(d_i^K)}{T} \tag{1}$$

Effective FA rate: It is illustrated in Equation 2.

$$R_1(K) = \frac{\sum_i (1 - c_f(d_i^K))}{T} \tag{2}$$

where T is the length of the audio processed, K is the term detected and $c_f(d_i^K)$ is the feature score for the detection i of the term K in equations 1 and 2.

The feature distribution represented in Fig.3 shows a very large overlapping between the two different classes (hit and FA).

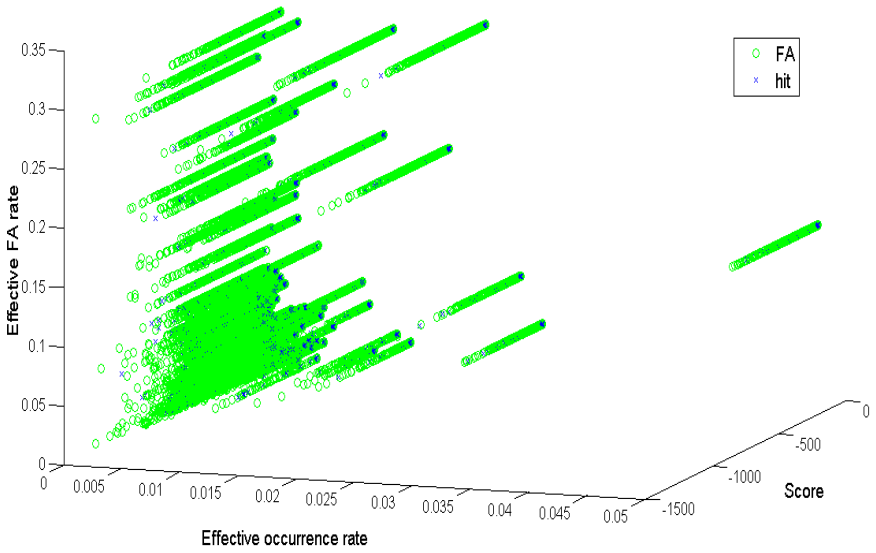

Fig. 3 Three dimensional graphic representation of the test set detections. For each detection (false alarm or hit), the three axis represent the three term-dependent features. Circles represent false alarms and crosses represent hits.

3.2 MLP-based classification

A 3-layer MLP was trained to build the MLP-based confidence measure from the features explained above. The structure of the MLP is composed of an input layer, a hidden layer and an output layer. The input layer has three units (in accordance with the number of features). The hidden layer has 5 hidden units and uses a sigmoid activation. The output layer has two units (hit and FA) and uses a softmax activation to achieve a reliable probability, which makes possible the final hit/FA classification. It must be noted that by using the softmax function, we force the sum of the output units to 1, thereby making the MLP a probabilistic classifier. The standard error backpropagation algorithm was used to train the model, as described in [3].

3.3 Evolutionary algorithm-based classification

In our evolutionary confidence measure we use EDA to minimize the number of classification errors and to classify a set of detections as hit or FA using its output projection. Since the number of errors is a discrete non-differentiable quantity, it has to resort to an optimization method such as an evolution strategy [2] to minimize it. The algorithm consists of the following steps:

- An initial random projection is generated.
- The projection's worth (fitness) is the number of errors committed by assigning training patterns to the class of the closest mean in the projected space.
- The following steps are repeated until a prescribed number of generations without improvements is reached:
  - A new projection is generated by adding independent normal perturbations to each component of the current projection.
  - The number of misclassified patterns is calculated for the new projection.
  - The new projection becomes the current one when the error is reduced.
- The current projection is output, and the data will be classified as belonging to the closest projected train mean.

4 Experiments

4.1 Experimental Setup

The input acoustic signal is sampled at 16kHz and stored with 16 bit precision. Mel Frequency Cepstral Coefficients (MFCCs) were computed at 10ms intervals within 25ms Hamming windows. Energy and first and second order derivatives were appended giving the 39 MFCCs used to represent the signal.

The set of 47 phones in Spanish language [9] has been used during the lattice generation. Hidden Markov Models (HMMs) were used as acoustic models to represent the set of phones and they were context-dependent with 8-components Gaussian Mixture Models (GMMs).

The Spanish Albayzin database [7], which contains two different sub-corpora, has been used in the experiments: a phonetic corpus and a geographic corpus. Each contains a training set and a test set. The training of the HMMs was made from the ***phonetic training set***. A bigram was used as language model during the phone recognition process (lattice generation). It was built from the ***phonetic training set*** as well. The number of components GMMs in the context-dependent acoustic models was tuned for phone accuracy on the ***phonetic test set***. The parameters *word insertion penalty* and *language scale factor*, used within the decoding process, were tuned on the ***geographic training set***. Finally, the ***geographic test set*** was used as test set for the system evaluation.

To train the MLP and EDA we have used the ***geographic training set*** and a list composed of 500 terms, which appeared 12651 times in this corpus to ensure an enough set of examplers for training. To solve the imbalance between classes (hits and FAs), we have duplicated the examplers of the minority class (hit) to equal the number of hits and FAs used to train both the MLP and EDA. The parameter tuning for the MLP used the same ***geographic training set*** but a different list of terms, composed of 105 terms, appearing 10423 times to ensure a reliable parameter estimation. In tuning the parameters for the MLP, the number of hidden units (5 in our case) was chosen to maximise the classification accuracy by cross-validation in the ***geographic training set***. We have used the original EDA parameters: an $(15, 100|2)$-ES with mutation step $\sigma = 0.15$ and 100 generations without improvements. Therefore, it was not necessary to make any tuning in this algorithm.

To test the system we have selected 400 terms appearing 11331 times in the ***geographic test set***. The list of terms used for training and parameter tuning had no overlapping between them. In order to simulate a real environment, the test set (400 terms in total) contained only 229 terms common to the training set.

4.2 Results and discussion

Results are presented using the miss ratio (%miss) and FA ratio (%FA) widely used for speech recognition and keyword spotting tasks. These values are defined as follows:

$$miss\ ratio = 1 - hits\ ratio \tag{3}$$

where *hits ratio* is defined as follows:

$$hits\ ratio = \frac{Number\ of\ occurrences\ correctly\ detected}{Number\ of\ actual\ occurrences\ in\ the\ audio} \tag{4}$$

$$FA\ ratio = \frac{Number\ of\ occurrences\ detected\ that\ do\ not\ appear\ in\ the\ audio}{Number\ of\ non\ relevant\ words\ in\ the\ audio} \quad (5)$$

Figure 4 represents the curves plotting these two values, in which the performance of each confidence measure for different operating points is presented. The curves are got by varying the threshold to remove those terms found by the lattice search algorithm whose score remains below it. Therefore, the standard output of the lattice search step can be considered as the confidence measure based on the score in this representation, and the MLP and EDA confidence measures also incorporate it to make possible the whole curve computation.

The three curves show the improvement got by the MLP and EDA-based classification techniques over the score-based confidence measure alone, as they produce the same number of FAs and more hits than the latter for every operating point when the threshold is used to compute the whole curve. It means that actually both MLP and EDA classify some detections as FA that the score-based confidence measure alone is not able to reject. It should be also considered that the end of the MLP and EDA curves represents the number of hits and FAs that remain in the system before applying any threshold (i.e., the direct output after the confidence measure) and at this point, both the MLP and EDA outperform the rate achieved by the score-based confidence measure. It is also shown the improvement achieved by the EDA confidence measure over the MLP-based classification when the FA is high and the contrary when the FA is low. It means that the MLP is able to reject FAs with a high score, and EDA keeps more hits with a low score. It should be considered that the EDA implementation makes use of a linear implementation in classifying the detections as hit or FA, and in Figure 3, it is seen that both the hits and FA have a very large overlapped margin. It may explain the worse performance of the EDA than the MLP when the FA is low.

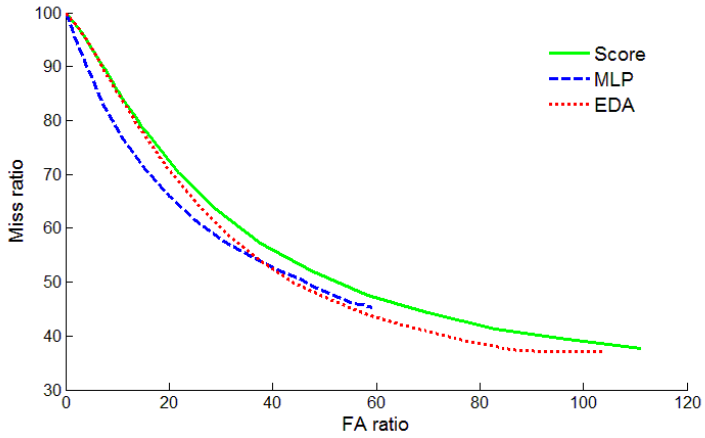

Fig. 4 Results in terms of *FA ratio* and *Miss ratio* for each confidence measure (Score, MLP and EDA) in the test set.

5 Conclusions

Confidence measures play a very important role in keyword spotting and those based on classification techniques have shown to be an important contribution to increase the system performance. We have presented an evolutionary algorithm integrated into the framework of the confidence measures for a keyword spotting system to reject those keywords that are classified as FA. We have compared its performance with an MLP and a score-based confidence measure and we have shown that the EDA outperforms the rate achieved by the score-based confidence measure for every operating point and performs better than the MLP when the FA is high.

Our evolutionary procedure can be improved in several ways. First of all, we can make use of the non-linear version of EDA. Besides, designing the fitness function to maximize the number of hits may keep more putative detections (therefore more hits) which, along with other new confidence measures, may lead to a better improvement in the keyword spotting system. We are currently working on this point and hope to report the results in the near future.

Acknowledgements Sponsored by CAM/UAM, project number CCG08-UAM/TIC-4428.

References

1. Ben Ayed, Y., Fohr, D., Haton, J. P., Chollet, G. (2002) Keyword spotting using support vector machines. Int. Conf. Text, Speech and Dialogue. 285–292
2. Beyer, H. G., Schwefel, H. P. (2002) Evolution Strategies. A Comprehensive Introduction. Natural Computing. 1:3–52
3. Bishop, C.M. (1995) Neural Networks for Pattern Recognition. Oxford University Press
4. Cuayahuitl, H., Serridge, B. (2002) Out-of-vocabulary Word Modeling and rejection for Spanish Keyword Spotting Systems. MICAI. 156–165
5. Goldberg, D. E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley
6. Kim, J.G., Jung, H.Y., Chung, H.Y. (2004) A Keyword Spotting approach based on Pseudo N-gram language model. SPECOM. 156–159
7. Moreno, A. et al (1993) Albayzin Speech Database: Design of the Phonetic Corpus. Eurospeech. 1:653–656
8. Ou, J., Chen, C., Li, Z. (2001) Hybrid neural-network/HMM approach for out-of-vocabulary words rejection in mandarin place name recognition. ICONIP
9. Quilis, A. (1998) El comentario fonológico y fonético de textos. ARCO/LIBROS S.A.
10. Sierra, A., Echeverría, A. (2006) Evolutionary Discriminant Analysis. IEEE Tran. Evo. Comp. 10(1):81–92
11. Szoke, I. et al (2005) Comparison of Keyword Spotting Approaches for Informal Continuous Speech. ICSLP. 633–636
12. Tejedor, J., Colás, J. (2006) Spanish keyword spotting system based on filler models, pseudo N-gram language model and a confidence measure. IV Jornadas en Tecnología del Habla. 255–260
13. Wang, D. et al (2009) Term-dependent confidence for Out-Of-Vocabulary Term Detection. Interspeech. 2139–2142
14. Young, S. et al (2002) The HTK Book V3.4. Microsoft Corp. and Cambridge Univ. Eng. Dep.