Universal approach for sequential audio pattern search

Róbert Gubka, Michal Kuba, Roman Jarina
Faculty of Electrical Engineering, University of Žilina
Univerzitná 1, 010 01 Žilina, Slovak Republic
Email: robert.gubka@fel.uniza.sk

Abstract—**This article deals with universal sequential audio pattern search and sound recognition method. Inspired by classical phoneme-based speech recognition and word spotting systems, where longer speech patterns are formed by sequences of basic speech units, we propose a methodology for creating a finite database of elementary sound models. These models can be arbitrary concatenated into sequences, thus forming a model of the required acoustical pattern or sound event.**

I. Introduction

AUTOMATIC speech recognition and word spotting systems are nowadays getting to the forefront in daily use. Intelligent human-computer communication interfaces allow us to take up the control over electronic equipment using our voice, making their usage more native and comfortable. All this is possible thanks to rigorous research in the field of human speech production and recognition. However, voice operated devices can only work with human speech and react only to specific spoken keywords or phrases in certain language, mostly English.

Universal intelligent system should be versatile, easily expandable for new commands in different languages, have ability to learn and operate also with non-speech sounds and acoustical events, e.g. for better evaluation of content and context of a situation. Such system can be adopted in different application areas. Audio is useful especially in situations when other sensors fails to reliably detect an event. For example in the context of surveillance systems, in weakly illuminated places, public transport areas, public halls or streets where it is not possible to evaluate video, or visual information alone is unreliable, the audio events spotting system can be very useful as burglar or violence alarm by detecting sound events, like glass brake, shout, footsteps, or any other significant sound defined by user [1], [2].

Another topic in the field of audio processing is information retrieval over multimedia content. Currently, huge amount of multimedia data such as music recordings, broadcast news, dialogs and conversations, etc., are available in large audio databases. However, most of these data are unstructured and have limited or no tag information about the content, hence, it is not easy for user to locate desired audio samples or segments.

Most of the existing applications perform the content-based analysis by segmentation of the audio data and subsequently classification of audio sequences into one of the specified sound classes, represented mostly by a statistical model [3]–[5]. Disadvantage of this approach is, that a sufficient amount of representative data is needed for each sound class to be created and user has only limited possibility to adding new queries into search. Moreover, with growing number of sound classes or queries in search also the computation and memory demands grow.

”Query-By-Example” (QBE) paradigm is an alternative approach to multimedia information retrieval. In the context of audio information, the user provides a short sound clip as a query and the system returns audio samples that are similar to the query. For example, the user provides a short utterance spoken by a particular person and expects that the system returns all the samples from the audio (video) database that contain the voice of the same person. Or the user gives a sample of an applause sound and the system should return all clips from the audio/video content that contain applause.QBE approach is also very popular in music information retrieval [6], [7]. The challenge of the QBE approach is that only very limited amount of training/reference data are available in advance and sound classes are apriori not known, thus conventional statistical model-based approaches and learning algorithms cannot be straightforward adopted.

Among various approaches to example-based audio event detection and retrieval, the most popular ones are based on similarity measure in audio feature vector space [8], [9] or hidden Markov models (HMM) [10], [11]. The approach in [11] was based on feature-based segmentation of audio using a dynamic Bayesian network. The inherent similarity or difference between sounds was determined by the corresponding similarity or difference between the audio features trajectory represented by HMM that approximates a general trend in query time behaviour.

In this paper we adopt different strategy to HMM modeling of an audio pattern. We propose the HMM approach that is inspired by methods very well explored in automatic speech recognition (ASR) and ASR-based spoken term retrieval. Every spoken word or sentence of speech of given language can be formed as a combination of basic acoustical-linguistic units, which are called phonemes. In every languages, there is a finite number of phonemes, e.g. there are approximately 44 phonemes in English language (may differ with particular dialect). In ASR-based keyword spotting systems, the search space is created only by statistical models of phonemes and

978-1-4673-4471-5/$25.00 © 2013, IEEE

keywords or speech patterns are added into search as logical sequences of these models.

Our effort was to adopt the concept of elementary units for general sound and language independent speech recognition task. This approach requires to define a basic unit of sound (hereinafter called the elementary sound) and create a finite, but sufficiently large database of these units. For this purpose, we have adopted methods of unsupervised cluster analysis applied over huge amount of various audio data to create a database of elementary sound models. Major contribution of our proposal is that the elementary sound models can be treated as analogy to phoneme-based models in speech processing. Thus, statistical methods for speech recognition can be applied for general audio. As it is known to the authors, no similar method has been proposed in literature.

II. Concept of Elementary Sound Units

A. Elementary sounds

The idea of elementary sound units is based on the assumption that, in general, any acoustical pattern can be synthesized as a concatenation of short-time sound elements, stored in finite (but sufficiently large) sound inventory. This idea is derived from speech recognition and keyword spotting systems based on concatenation of sub-word phoneme units from finite database, to represent the sentences of speech. These units are predominantly represented by their parametric statistical models, created and estimated according to corresponding acoustical training examples.

Of course there are many challenges to successfully implement such approach. The biggest problem is the fact that unlike speech, which can be easily modeled since its physical production is known, general sounds are produced by unlimited number of ways and thus can be of infinite variance in temporal-spectral behavior. Unlike the phonemes, we can not practically create a database of the acoustical examples for general sound basic units. Instead, we have performed a cluster analysis over statistical models and created a database of the elementary sound models. The process of the creation is described below.

B. Parametric modeling

First step to create the database of the elementary sound models is to select a suitable method for statistical modeling and model parameters. For this purpose, we choose modeling via hidden Markov models (HMM), which is suitable for statistical description of time series of observations and therefore is commonly used in speech and general sound recognition tasks. A continuous-density HMM with N states consists of a set of parameters that generally comprises the matrix of transition probabilities, the initial state distribution, and the parameters of the state output density function, which is mainly approximated by a mixture of Gaussian components that can be expressed as follows:

$$P(\boldsymbol{x}|S) = \sum_k w_k \mathcal{N}_k(\boldsymbol{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where $\mathcal{N}_k$ is the Gaussian component with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, and w_k is the weighting coefficient of this component in the state of the model. However, as the number of states in the search space grows, the system becomes more computationally and memory demanding. For this reason, we adopt a semi-continuous-density models, where all states share the same Gaussian components. Output density function of particular state is then determined only by the weighting vector $\boldsymbol{w}$.

Furthermore, assuming that only one of the components has major contribution to the resulting likelihood of the state, the summation operation in (1) can be replaced by the selection of maximal value. This modification reduce the processing time with minimal impact on resulting likelihood (mean difference less than 5%).

Similar to phoneme models, we choose a 3-state left-to-right model structure with equal transition probabilities, which can therefore be omitted.

C. Unsupervised clustering

One of the problems of statistical modeling is the determination of optimal number of Gaussian mixture components for output density function, which is usually determined by experiments [3], [4]. More sophisticated approach is based on Bayesian or Akaike Information Criterion [12], [13], Kullback-Leibler divergence [13], and unsupervised clustering methods [14], [15]. In [15], unsupervised *K-Variable K-means* clustering algorithm was proposed. This algorithm was adopted in our work to determine the optimal number of mixture components and also for derivation of the elementary sound models. The clustering algorithm is described below.

Algorithm 1 K-Variable K-means

```
Compute $m$ and $s$ as the mean and standard deviation of the distances between any pair of frames.
Set threshold distances: $T_1 = m - C.s$; $T_2 = m + C.s$ where $C \in < 0.5; 1.5 >$
Create 1st centroid as $\boldsymbol{c}_1 = \arg\max_i(\|\boldsymbol{x}_i\|)$
for $\forall \boldsymbol{x}_i$ do
    $d_i = \min_j(d_{ij}(\boldsymbol{x}_i, \boldsymbol{c}_j))$; Find the distance to the closest centroid.
    if $d_i < T_1$, Make $\boldsymbol{x}_i$ member of cluster $j$;
    if $d_i > T_2$, Make $\boldsymbol{x}_i$ the center of a new cluster;
end for
Make all the remaining unclassified vectors members of their closest cluster.
```

D. Database of the Elementary Sound Models

The most important part of our system for general audio pattern searching is the database of the elementary sound models. Due to the gargantuan number of different sounds that may generally occur, it is impossible to create a finite database of acoustical training examples for elementary sound models. Therefore we have adopted the cluster analysis do derive and group sounds with similar statistical characteristics.

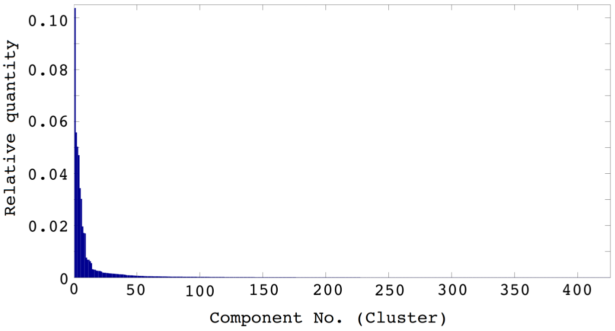

Fig. 1. Distribution of acoustical observations among clusters.

First, a sound database was collected, that consist of more than 30 hours of various short audio clips. This database involve different types of environmental and machinery sounds and noises, animal sound, human produced sounds and speech of different languages, music of different genres, etc.

The database was processed and the audio features described in section III were extracted. To obtain the Gaussian mixture components, all feature vectors were divided into clusters. Before the actual clustering, confusing data vectors were eliminated. The centroid of the whole data set was computed and the mean m and standard deviation s of distances of data vectors from this centroid were computed. Vectors with distance $d > m + 3.s$ were discarded.

Remaining data vectors were clustered using the unsupervised clustering and the means $\boldsymbol{\mu}_k$ and diagonal covariance matrices $\boldsymbol{\Sigma}_k$ occurred in (1) were computed for each cluster respectively. Clusters with less than 30 observations were discarded. As the result of clustering, state output density function consist of 426 Gaussian mixture components. Fig. 1 shows the distribution of acoustical observations among clusters.

Next, audio stream from all sound clips was formed and divided into 1 second long segments with 0.5 second overlap. From each segment a 3-state model was estimated. Because the models are defined by their weighting coefficient vector in each state, estimated models were clustered by unsupervised clustering. A histogram of distribution of the models within the clusters was computed. The clusters were assorted from the biggest to the smallest according to the number of members within the cluster. The elementary sound models were then derived as the means of the clusters. The clustering results in more than 10^4 clusters, although most of them comprise only one member. Therefore, only first 500 models were adopted for experiments.

Each of these models describe the audio segment in that the acoustical observations are statistically very similar. Conversely, acoustical observations of different elementary sounds differ in their characteristics.

III. Audio Features Selection

In order to achieve the best performance for classification, we have selected features that can capture the temporal and spectral characteristics of audio. Following work in [16], in which the features were selected by optimization algorithm, we have selected the following features:

1) *Line spectral frequencies/pairs (LSF/LSP)* - used as an alternative to linear prediction coefficients. The LSF are obtained by decomposing the LP filter transfer function A(z) into pair of auxiliary polynomials:

$$\begin{aligned} P(z) &= A(z) + z^{-p+1}A(z^{-1}) \\ Q(z) &= A(z) - z^{-p+1}A(z^{-1}) \end{aligned} \tag{2}$$

where $P(z)$ is symmetrical and $Q(z)$ asymmetrical $p+1$-order polynomial, where the zeros of $A(z)$ are mapped onto the unit circle in the z–plane.

2) *Spectral flux (SFX)* - measures changes in the shape of magnitude spectrum by calculating the difference between magnitude spectra of successive frames. The spectral flux is computed for frame at discrete time t as follows:

$$SFX(t) = \frac{\sum_k [a_k(t) - a_k(t-1)]^2}{\sqrt{\sum_k a_k(t)^2}\sqrt{\sum_k a_k(t-1)^2}} \tag{3}$$

where a_k is the k-th element of magnitude spectrum of given frame. Spectral flux describes the temporal changes of magnitude spectrum, thus represents the dynamic coefficients of spectrum.

3) *Zero crossing rate (ZCR)* - the number of time-domain zero-crossings within a frame, computed as a number of sample sign changes.

These features were extracted using the Yaafe [17] extraction tool. The resulting feature vector consists of 10 LSF's, one SFX, and one ZCR coefficient.

IV. Audio Pattern Searching

The theoretical background for sequential audio pattern searching was taken from [18] where a decoder for keyword spotting was proposed. Its function is based on Viterbi algorithm with propagation of accumulated score through the

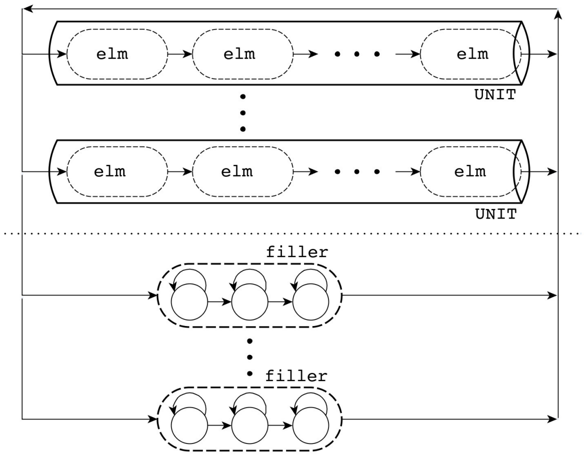

Fig. 2. Looped network of units and fillers.

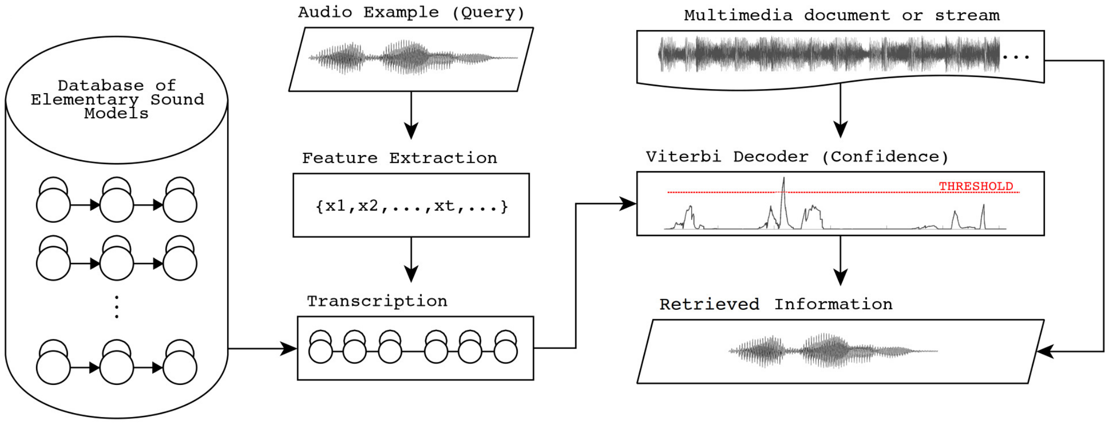

Fig. 3. Query-by-Example search.

looped network of units that represents searched keywords, and fillers.

In our implementation, the basic unit of the decoder is the elementary sound. In process of creating the model of demanded audio pattern, a representative audio sample is passed into decoder and transcribed into sequence of the elementary sound models. This transcription is found as the path through the elementary sound models with the highest score achieved for the training example. This logical sequence of elementary sounds is then added into search space as a unit, representing the searched audio pattern. Example of model network with units and fillers is shown in Fig.2.

In the process of decoding, each acoustical observation must be assigned to one of the states in model network. The acoustical observations between segments that correspond with searched patterns are assigned to the filler models, that represents any sound that may occur in background. This definition of filler offers us the possibility of using the proposed database of elementary sound models itself.

Another benefit of using the elementary sounds as fillers is that we can compute the confidence of particular unit as proposed in [18]. The confidence $C(u,t)$ of unit u at discrete time t is defined as normalized acoustic score as follows:

$$C(u,t) = S(u,t)/S(f_c,t) \tag{4}$$

where $S(u,t)$ it the acoustic score achieved for segment of audio by unit u, and $S(fc,t)$ is the acoustic score achieved by the best concatenation of fillers (in our case the elementary sound models) for the same segment. It follows that the confidence reaches the maximum value of 1 only when the score of the unit and the score of fillers concatenation are equal. In this case, the audio segment precisely correspond with the unit training example. In other cases when $C(u,t) < 1$ the probability of correct detection decreases and proper threshold must be set for experimental data. Fig. 3 shows the principal functions of the complex system for query-by-example general audio pattern recognition.

V. Experimental Results

A. Experiment setup

The experiments on the proposed database of the elementary sound models were performed in task of audio pattern search in recordings, which include acoustical patterns of five different sound types: *applause, crying, laughing, gunshot and speech (10 Slovak keywords).* Ten artificial audio tracks were created by random concatenation of 20 audio examples from each sound class and various types of environmental background noises respectively. For each audio track, the examples were chosen randomly from the sound database.

Accuracy of the search was evaluated on the level of acoustical observations against the human annotation of records with common precision (P), recall (R), and F_1-measure (F_1) metrics defined as follows:

$$P = \frac{n_{correct}}{n_{total}}; R = \frac{n_{correct}}{n_{target}}; F_1 = \frac{2.P.R}{P+R}; \tag{5}$$

where $n_{correct}$ stands for correct positive detections, n_{total} for total positive detections and n_{target} for target positive detections.

B. Audio event detection

For each searched sound example, the unit was created, as described in Section IV. Each of these units was used as query for searching. Units were able to find corresponding training examples with practically 100 % accuracy and confidence close to 1. Although using simple correlation will be much more efficient in this case, this experiment proved our prior assumption that specific audio pattern can be modeled as a sequence of the elementary sound models.

In the next experiment, only one representative example for each class was used as query for search. By changing the confidence threshold, the balance between precision and recall was set, to obtain highest possible F-measure score. The decoder was able to find also other audio segments similar to the queries. Overall average precision and recall reaches 62.5 % and 58.5 % respectively.

TABLE I
EXPERIMENTAL RESULTS ON VARIOUS ACOUSTIC PATTERNS

Pattern (Class)	*	**Number of examples in query**		
		1	***2***	***3***
Applause	P	73.40 %	81.55 %	89.33 %
	R	76.78 %	97.81 %	96.75 %
	F_1	75.60 %	88.95 %	92.89 %
Crying	P	55.69 %	85.98 %	81.52 %
	R	55.50 %	52.49 %	57.71 %
	F_1	55.60 %	65.18 %	67.58 %
Laughing	P	52.88 %	98.73 %	83.76 %
	R	41.21 %	52.07 %	80.08 %
	F_1	46.32 %	68.18 %	81.88 %
Gunshot	P	87.18 %	84.35 %	91.00 %
	R	80.68 %	79.25 %	88.62 %
	F_1	78.07 %	81.72 %	89.80 %
Keywords	P	43.17 %	75.41 %	80.78 %
	R	48.38 %	63.17 %	70.22 %
	F_1	45.63 %	68.75%	75.13 %

In the next search run, one additional representative example of each class was added into the search space, so that two examples were in the query. By setting the confidence threshold for each example of representative pair, the overall average precision and recall increased to 85.2 % and 69.0 % respectively. Lastly, three representative examples were selected as queries and put into search. The confidence threshold was set for each examples of three. The average precision and recall again increased to 85.3 % and 78.7 % respectively.

The advantage of our system is that if a user has only one example of demanded audio pattern (query-by-example), new examples can be added into search as selected audio segments found in previous run. Thus, the system has ability to ”learn” from users feedback after the search. Table I shows the average results achieved using 1, 2, and 3 examples in query for each sound class.

VI. Conclusion

The system for universal sequential audio pattern search has been proposed. Statistical model specifications were introduced and the database of elementary sound models was created, using the unsupervised clustering method. Experimental results show that it is possible to adopt the concept of elementary sound units for general audio pattern modeling and recognition. However, a proper confidence threshold must be set experimentally for each unit to obtain the best possible result. Experiments also show that adding more examples of particular audio pattern can significantly improve searching results. If these examples are selected by a user from previous search run results, the system is also able to learn according to user feedback.

In our future work, we aim to include expectation–maximization algorithm in the clustering and adopt Viterbi alignment and discriminative training for better discrimination of elementary sound models. We also plan to compare the performance of the system on larger set of audio features extracted from audio data.

Acknowledgment

This work has been supported by the Centre of excellence for systems and services of intelligent transport II., ITMS 26220120050 supported by the Research & Development Operational Programme funded by the ERDF.

References

[1] Rouas, J. L., Louradour, J., & Ambellouis, S. (2006, September). Audio events detection in public transport vehicle. In Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE , Sept. 2006, pp. 733-738

[2] Vozarikova E.; Pleva, M.; Juhar, J.; Cizmar, A.,”Surveillance system based on the acoustic events detection”, Journal of Electrical and Electronics Engineering. Vol. 4, no. 1, 2011, pp. 255-258.

[3] Lian-Hong Cai; Lie Lu; Hanjalic, A.; Hong-Jiang Zhang; Lian-Hong Cai, ”A flexible framework for key audio effects detection and auditory context inference,” IEEE Transactions on Audio, Speech, and Language Processing, vol.14, no.3, pp.1026,1039, May 2006

[4] Heittola, T.; Mesaros, A.; Eronen, A.; Virtanen, T.; ”Audio context recognition using audio event histograms,” 18th European Signal Proc. Conf. (EUSIPCO-2010) Aalborg, Denmark, August 23-27, 2010

[5] Atrey, P.K.; Maddage, M.C.; Kankanhalli, M.S., ”Audio Based Event Detection for Multimedia Surveillance,” IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 , vol.5, no., pp.V,V, 14-19 May 2006

[6] Lemström, K., Tzanetakis, G.: Music Information Retrieval. In Bates, M.J. (Ed.): Understanding Information Retrieval Systems: Management, Types, and Standards, CRC Press, 2012.

[7] Chandrasekhar, V., Sharifi, M., & Ross, D. A. ”Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications.” In Int. conf. ISMIR, pp. 801-806. 2011.

[8] Stan Z. Li,”Content-Based Audio Classificafion and Retrieval Using the Naerest Feature Line Method”,IEEE Transactions speech and audio processing, vo1.8, No.5, Sept. 2000.

[9] Marko Helen, Tuomas Virtanen . Audio query by example using similarity measures between probability density functions of features. EURASIP Journal on Audio, Speech, and Music Processing, 2010.

[10] Velivelli, A.; Zhai, C.X.; Huang, T.S., ”Audio segment retrieval using a short duration example query,” IEEE Int. Conf. on Multimedia and Expo, ICME '04., vol.3, pp.1603-1606, June 2004

[11] Wichern, G., Xue, J., Thornburg, H., Mechtley, B., & Spanias, A. Segmentation, indexing, and retrieval for environmental and natural sounds., IEEE Transactions on Audio, Speech, and Language Processing 18(3), 2010, pp. 688-707

[12] Tenmoto, H.; Kudo, M.; Shimbo, M., ”Determination of the number of components based on class separability in mixture-based classifiers,” Third International Conference Knowledge-Based Intelligent Information Engineering Systems, 1999. vol., no., pp.439,442, Dec 1999

[13] Xiao-Bing Li; Ren-Hua Wang, ”State Divergence-Based Determination of The Number of Gaussian Components of Each State in HMM,” Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2006, vol. 1, no., pp.I,I, 14-19 May 2006

[14] Mucciardi, Anthony N.; Gose, Earl E., ”An Automatic Clustering Algorithm and Its Properties in High-Dimensional Spaces,” IEEE Trans. on Systems, Man and Cyber., vol.SMC-2, no.2, pp.247,254, April 1972

[15] Reyes-Gomez, M. J.; Ellis, D. P. W., ”Selection, parameter estimation, and discriminative training of hidden Markov models for general audio modeling,” International Conference on Multimedia and Expo, 2003. ICME '03. vol.1, no., pp.I,73-6 vol.1, 6-9 July 2003

[16] Chmulik, M.; Jarina, R., ”Bio-inspired optimization of acoustic features for generic sound recognition,” 19th Int. Conf. on Systems, Signals and Image Proc. (IWSSIP), 2012, pp.629,632, 11-13 April 2012

[17] B.Mathieu; S.Essid; T.Fillon; J.Prado; G.Richard; YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software, proceedings of the 11th ISMIR conference, Utrecht, Netherlands, 2010.

[18] J. Nouza, J. Silovsky, ”Fast keyword spotting in telephone speech,” Radioengeneering 2009, vol. 18, no. 4, Dec. 2009, pp. 665-670