Multimed Tools Appl (2013) 62:35–49
DOI 10.1007/s11042-011-0914-y

Automatic detection of known advertisements in radio broadcast with data-driven ALISP transcriptions

Houssemeddine Khemiri · Gérard Chollet · Dijana Petrovska-Delacrétaz

Published online: 19 November 2011

Abstract This paper presents an audio indexing system to search for known advertisements in radio broadcast streams, using automatically acquired segmental units. These units, called Automatic Language Independent Speech Processing (ALISP) units, are acquired using temporal decomposition and vector quantization and modeled by Hidden Markov Models (HMMs). To detect commercials, ALISP transcriptions of reference advertisements are compared to the transcriptions of the test radio stream using the Levenshtein distance. The system is described and evaluated on one day broadcast audio streams from 11 French radio stations containing 2070 advertisements. With a set of 2,172 reference advertisements we achieve a mean precision rate of 99% with the corresponding recall value of 96%. Moreover, this system allowed us to detect some annotation errors.

Keywords ALISP tools · Advertisement detection · Copy detection · Data-driven speech segmentation · HMM models

H. Khemiri (✉)
Department of Signal and Image Processing, TELECOM ParisTech
and TELECOM SudParis, 37-39, rue Dareau, 75014 Paris, France
e-mail: khemiri@telecom-paristech.fr

G. Chollet
Department of Signal and Image Processing, TELECOM ParisTech,
37-39, rue Dareau, 75014 Paris, France
e-mail: chollet@telecom-paristech.fr

D. Petrovska-Delacrétaz
Electronics and Physics Department, TELECOM SudParis,
9, rue Charles Fourier, 91011 Évry, France
e-mail: dijana.petrovska@it-sudparis.eu

1 Introduction

Automatic audio indexing has become a major concern today. Given the increasing amount of audio data, new indexing applications (music recognition, advertisement detection, radio broadcast monitoring, jingle detection, etc) are constantly developed. We are interested in detecting known commercials in a continuous radio stream for media metric purposes (count the number of times each advertisement was broadcasted).

In the literature the majority of proposed systems to detect advertisements rely on the same underlying concept: audio fingerprinting [8, 12, 16, 20]. This technique involves the extraction of a signature (or fingerprint) for each audio document stored in a reference database. An incoming audio excerpt is identified by comparing its fingerprint with those of the reference database.

Usually the fingerprint algorithm consists in performing a spectral analysis in order to produce an efficient representation of the audio signal [7]. A number of methods are based on the analysis of the energy in subbands (such as, for example, the sign of the energy differences [15], or the spectral magnitudes [22]). Other authors have also proposed more specific representations based on MPEG7 descriptors [1] or building constellation of spectral peaks, as in the *Shazam* solution [24].

Most of these approaches are capable of very high accuracies but suffer at varying degrees from degradations such as additive noise, compression or speed changes.

Few studies has so far focused on indexing techniques based on speech recognition methods, that are based on transforming the audio data in a sequence of phone units [6]. The set of previously trained models, i.e. phones can be used to segment speech data into a sequence of recognized units . During the training (learning) phase of the phone models, speech data along with its textual transcriptions are needed. Furthermore dictionaries and language models are required to transform the sequence of phones in a meaningful textual output. If the semantic meaning of the speech data is not needed, data-driven techniques can be applied in order to learn automatically a set of pseudo-phonetic models. The particularity of such methods is that no textual transcriptions are needed during the learning step, and only raw audio data is sufficient. In such a way any input audio data is transformed into a sequence of arbitrary symbols. These symbols can be used for indexing purposes.

This is the originality of the ALISP approach. ALISP is a data-driven technique that was first developed for very low bit-rate speech coding [10], and then successfully adapted for other tasks such as speaker [14] and language recognition [11]. Schwarz [23] showed that data-driven approaches are also applicable for music signals.

The advantage of ALISP tools is that no manual transcriptions are needed in the training phase and they can be easily deployed on new data, tasks, or languages.

In this paper we propose to use automatically acquired segmental units provided by ALISP tools to search for known advertisements in radio broadcasted streams. This task involves detecting and locating occurrences of an entire advertisement in audio streams. In this sense ALISP transcriptions of manually segmented reference advertisements are computed using HMM models provided by ALISP tools and stored in a reference database. Then reference transcriptions are compared to transcriptions of the radio stream using the Levenshtein distance. The proposed system is depicted in Fig. 1.

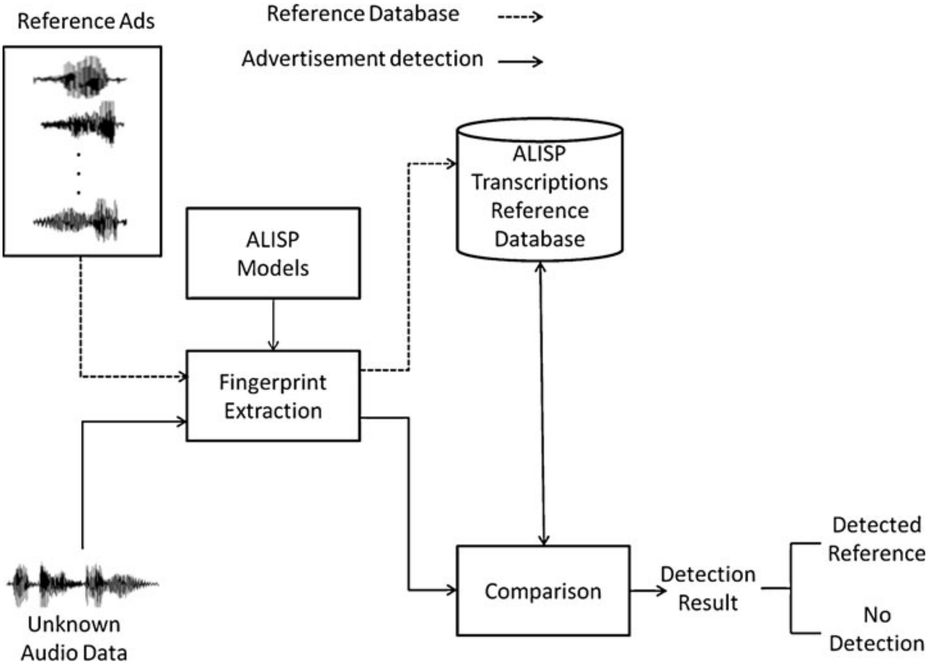


Fig. 1 Proposed ALISP-based system for audio fingerprinting

The paper is organized as follows. In Section 2, each module of the proposed system is presented. The radio broadcast database and experimental setup are described in Section 3. The experimental results are presented in Section 4. Conclusions and perspectives are given in Section 5.

2 Modules of the proposed ALISP-based system

Our system is based on automatically acquired segmental units provided by ALISP tools to search for known advertisements (denoted also as reference advertisements) in radio broadcast streams. In this sense ALISP transcriptions of known advertisements are computed using HMM models provided by ALISP tools and Viterbi algorithm and compared to transcriptions of the radio stream using the Levenshtein distance.

2.1 The acquisition and modeling of ALISP units

As explained in [9, 11, 13, 14, 21], and shown in Fig. 2, the set of ALISP units is automatically acquired through parametrization, temporal decomposition, vector quantization, and Hidden Markov Modeling.

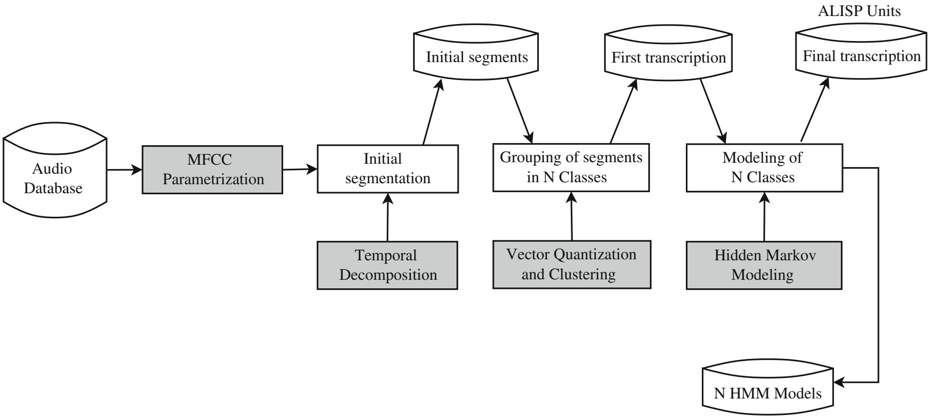


Fig. 2 Automatic Language Independent Speech Processing (ALISP) units acquisition and their HMM modeling

2.1.1 Parametrization

The parametrization of audio data is done with Mel Frequency Cepstral Coefficients (MFCC), calculated on 20 ms windows, with a 10 ms shift. For each frame, Hamming window is applied and a cepstral vector of dimension 15 is computed and appended with first and second order deltas.

2.1.2 Temporal decomposition

After the parametrization step temporal decomposition is used to obtain an initial segmentation of the audio data into quasi-stationary segments. This method was introduced originally by Atal [3] as nonuniform sampling and interpolation procedure for efficient parameter coding. The detailed algorithm to find interpolation functions can be found in [4]. Once interpolation functions are computed, their intersections are used to determine segment boundaries. The speech segments correspond at this point to spectrally stable portions of the signal. These segments are further clustered using Vector Quantization.

2.1.3 Vector quantization

The next step in the ALISP process is the unsupervised clustering procedure performed via Vector Quantization [19]. This method maps the P-dimensional vector of each segment provided by the temporal decomposition into a finite set of L vectors $Y = \{y_i; 1 \leq i \leq L\}$. Each vector y_i is called a code vector or a codeword and the set of all codewords is called a codebook. The codebook size L defines the number of ALISP units.

The codebook training is performed using vectors located in gravity centers of segments computed with temporal decomposition (one vector per segment). This training is done by a K-means algorithm with binary splitting. The initial labeling of the entire audio segments is achieved by assigning segments to classes using minimization of the cumulated distances of all the vectors from the audio segment to the nearest centroid of the codebook. This method, called Linde-Buzo-Gray (LBG) [18], results in a codebook size which is a power of 2. The result of this step is an initial segmentation and labeling of the training corpus. Refinement of these segments is done with Hidden Markov Modeling.

2.1.4 Hidden Markov Modeling

The final step is performed with the Hidden Markov Modeling procedure. The objective here is to train robust models of ALISP units on the basis of the initial segments resulting from the Temporal Decomposition and Vector Quantization steps. HMMs training is performed using the HTK toolkit [5]. It is mainly based on Baum-Welch re-estimations and on an iterative procedure of refinement of the models that is summarized as follows:

- Initialization of parameters: this step provides initial estimates for the parameters of HMMs using a set of observation sequences. First, a prototype HMM definition must be specified in order to fix the model topology. In this system, each ALISP unit is modeled by a left-right HMM having three emitting states with no skips. Covariances are diagonal, and computed for each mixture. The initialization of models is performed via HInit tool.
- Context independent re-estimation: the initial parameter values computed by HInit are then further reestimated by HRest tool using the Baum-Welch reestimation procedure. In the contrary of HInit in which each observation vector is assigned to a unique state, HRest assigns each observation to every state in proportion to the probability of the model being in that state when the vector was observed.
- Context dependent re-estimation: this re-estimation step uses the same Baum-Welch procedure as for the context independent re-estimation but rather than training each model individually all models are trained in parallel. This re-estimation is done by HERest tool. For each training utterance, the corresponding segment models are concatenated to construct a composite HMM which spans the whole utterance. This composite HMM is made by concatenating instances of the ALISP classes HMMs corresponding to each label in the transcription. The forward-backward algorithm is then used to accumulate, for each HMM in the sequence, the statistics of state occupation, means, variances, etc. When all of the training data has been processed, the accumulated statistics are used to compute re-estimates of the HMM parameters. It is important to emphasize that in this process, the transcriptions are only needed to identify the sequence of labels in each utterance. No segment boundary information is needed.
- Model refinement: this step consists in an iterative refinement of these HMMs by successive segmentation of the training data followed by re-estimations of parameters. The segmentation is performed using the HVite tool which is based on the Viterbi algorithm [25]. HVite matches an audio file against a network of HMMs and outputs its transcription.

2.2 Improving the ALISP models

This part is related to the adaptation and improvement of ALISP tools with regard to the task and the database used for known advertisements detection. In the initial system one Gaussian per state was used. The improvements concern mainly the number of Gaussian mixtures in each state of the HMM models.

In this part, a dynamic split of the states mixtures during the HMM modeling step of the ALISP units is introduced [13]. It is known that mono-Gaussians are not sufficient to model phone or pseudo-phone units and consequently mixtures of Gaussians are usually used. The main issue to be addressed when mixtures of Gaussians are used is the number of Gaussians. This number has to be chosen in order to allow precise modeling of ALISP units according to the quantity of data. A convenient compromise between model complexity and quantity of training samples needs to be found.

Therefore, a dynamic method is used to fix the number of Gaussians of each ALISP model. This method is based on the difference of recognition likelihoods. The number of Gaussians is increased iteratively in the models and after re-estimating the models the difference of recognition likelihoods at iteration m with those at iteration $m-1$ is computed. If this difference is below a certain threshold for a given ALISP unit model, the split of the states mixtures is stopped, otherwise the procedure continues.

2.3 Recognition of ALISP units

Given an observation sequence of features $Y = y_1, ..., y_T$, the recognized ALISP sequence is the one which is the most likely to have generated the observed data Y. In such a way any audio stream is transformed into a sequence of ALISP symbols. An efficient way to solve this problem is to use the Viterbi algorithm [25].

2.4 Similarity measure and searching method

After the recognition of ALISP units, the next important part of the proposed system is the matching process. This module compares the ALISP sequences extracted from observed audio signal against reference ALISP transcriptions stored in the reference database. First, the transcriptions of each reference advertisement (the ones that we are going to look for in the newly incoming audio stream) into a sequence of reference ALISP symbols has to be done. Then the test audio stream is transformed into a sequence of ALISP symbols. Once the ALISP transcriptions of reference and test data are done, we can proceed to the matching step.

The similarity measure used to compare ALISP transcriptions is the Levenshtein distance [17]. The Levenshtein distance is a special case of an edit distance. The edit distance between two strings of characters is the number of operations required to transform one of them into the other. When edit operations are limited to insertion, deletion and substitution this distance is called Levenshtein distance. At this stage the matching component used in our system is very elementary. In each step we move on by one ALISP unit in the test stream and Levenshtein distance is computed between reference advertisement transcription and the transcription of the selected excerpt from the audio stream. At the point when the Levenshtein distance is below a predefined threshold it means that we have an overlap with the reference. Then we continue the Levenshtein distance comparison by stepping on by one ALISP symbol until the Levenshtein distance increases relatively to its value in the previous step. This point indicates the optimal match, where the entire reference has been detected.

3 Radio broadcast database and experimental setup

The first part of this section deals with the radio broadcast database used to train ALISP models and to test our system. In the second part the evaluation protocol and experimental setup are described.

3.1 Radio broadcast database

YACAST (http://www.yacast.fr) has provided us with 26 days of broadcast audio of 13 French radio stations with their annotations. These records contain 2,172 different commercials that are repeated between 2 and 117 times. The mean duration of these commercials is 24 s and their total number in 26 days of recording is 14,953.

YACAST has done an initial manual annotation which is needed to define the experimental protocols and serve us also as a ground truth during the evaluation task. Annotations of advertisements are in XML format. They provide information related to the product name and ID, the radio ID, the commercial ID and the broadcast time.

3.2 Experimental setup

For the current experiment, only a part of the available data is used, corresponding to seven days of radio broadcast. The rest of the data will be exploited in future experiments. The seven days are split in different parts as follows:

- Development database: to train the ALISP models, 1 day of audio stream from 12 radios is used (leading to 288 h); three days of audio stream are used to study the stability of ALISP transcriptions of advertisements and to set the decision threshold for the Levenshtein distance.
- Reference database: it contains 2,172 advertisements and represents commercials to be detected in the radio stream. These advertisements correspond to audio segments previously broadcasted and manually annotated and extracted. The radio stream from whom a given reference was extracted corresponds to the first occurrence of the audio element in the database, and is not part of the evaluation set.
- Evaluation database: our system is evaluated on three days of audio stream from 11 radios (the equivalent of 33 days). This database contains 453 different commercials that are repeated between 1 and 12 times with 323 advertisements which are repeated more than once. The total number of advertisements present in the evaluation database is 2,070. Table 1 shows the number of advertisements present in each radio. Figure 3 shows the duration distribution of advertisements in the evaluation database.

The parametrization is done with the HTK toolkit v 3.2.1 [5], with the parameters defined in Section 2. For the vector quantization, the codebook size is 64.

For the multi-Gaussian HMM models the number of Gaussian components used per mixture obtained after the dynamic splitting stage is shown in Fig. 4. The mean value of Gaussian components per mixture is 6.

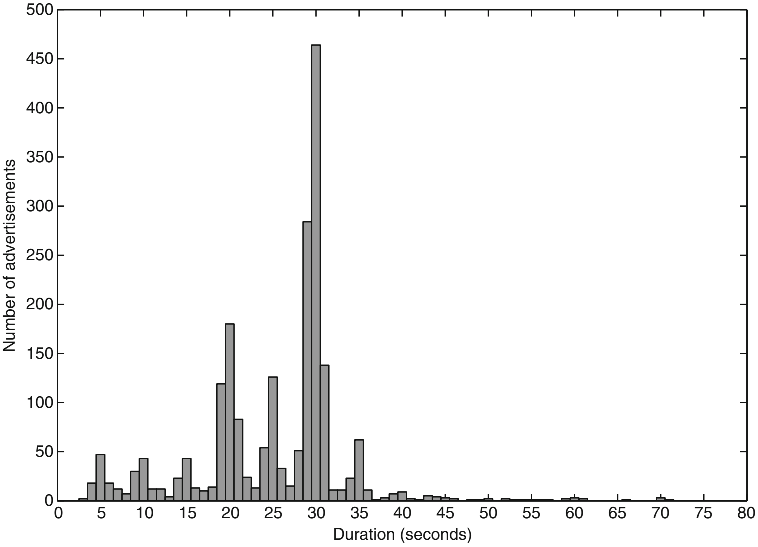


Fig. 3 Duration distribution of the 2,070 advertisements present in the evaluation database

4 Results

To detect commercials in a radio stream, ALISP transcriptions of reference advertisements are compared to the transcriptions of the radio broadcast using the Levenshtein distance. But before detecting commercials, the stability of ALISP transcriptions of these spots is studied. This study allowed us to fix the decision threshold of the Levenshtein distance on the development data set.

4.1 Characteristics of the ALISP units

The actual number of ALISP units is 65 (64+Silence model) and the average length per model is around 100 ms. Compared to the audioDNA indexing method described in [6] which extracts 800 gens per minute, ALISP tools give also a very compact way to represent audio data with 600 ALISP units per minute.

Table 1 Number of advertisements present in each radio stream in the evaluation (test) database set

Radio Id	Number of advertisements
1	178
165038	89
3	223
541	71
547	193
548	263
552	191
553	162
555	92
557	289
558	319

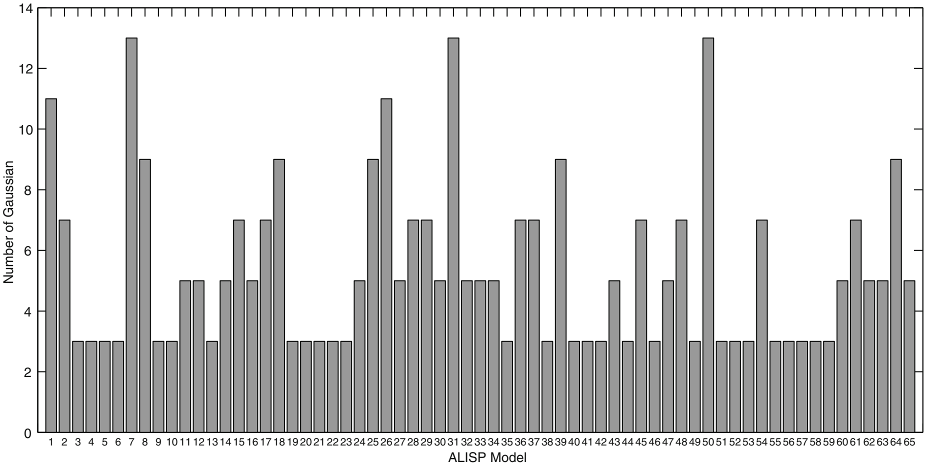


Fig. 4 Number of Gaussian components per mixture for the multi-Gaussian HMM ALISP models

Figure 4 shows a spectrogram of excerpt from a reference advertisement and two spectrograms of the same advertisement from two different radios with their ALISP transcriptions with the mono-Gaussian HMM model. Note the presence of some differences between ALISP transcriptions of the three advertisements. These

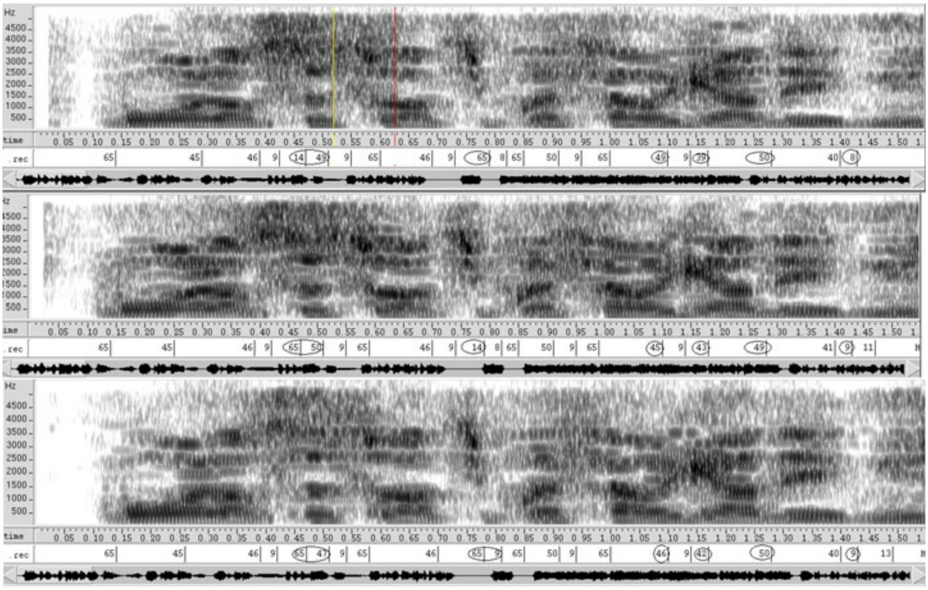


Fig. 5 Advertisement spectrograms with their ALISP transcriptions: first spectrogram is an excerpt from the reference advertisement, second one represent the same excerpt from radio 558 and the last one represents the same excerpt from radio 1. The Levenshtein distances between each of the two advertisements with the reference one are 0.26 and 0.24

differences could be explained by the similarity between some ALISP classes which leads to confusion during the recognition of these classes. The Levenshtein distance between each of the two advertisements with the reference in the Fig. 5 is 0.26 and 0.24.

4.2 Threshold setting

To study the stability of ALISP transcriptions and determine the decision threshold, two experiments were realized with mono and multi-Gaussian ALISP models:

- Compare ALISP transcriptions of the reference advertisements to the commercials in the radio recording (intra-pub experience).
- Compare ALISP transcriptions of reference advertisements to data that does not contain advertisements (extra-pub experience).

Figure 6 shows the distribution of the Levenshtein distance between ALISP transcriptions of references and advertisements in the radio recordings (denoted

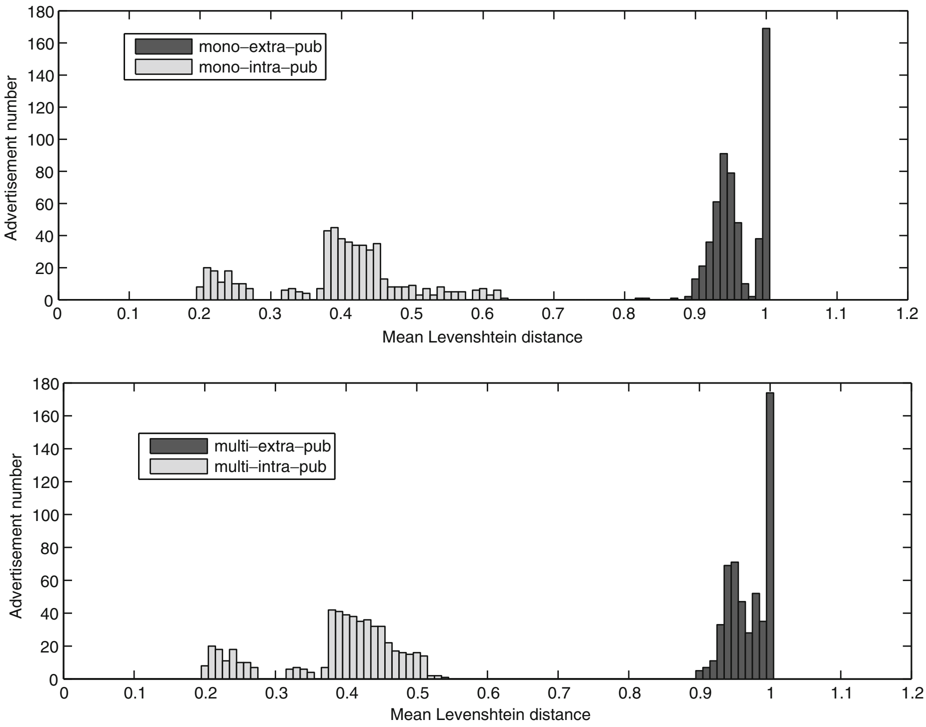


Fig. 6 Distribution of the Levenshtein distance between ALISP transcriptions of references and advertisements in the development radio recordings (denoted as intra-pub) and distribution of the Levenshtein distance between ALISP transcriptions of references and data that do not contain advertisements (denoted as extra-pub). Upper the Levenshtein distribution for the mono-Gaussian models and lower for the multi-Gaussian models

as mono-intra-pub and multi-intra-pub) and the distribution of the Levenshtein distance between ALISP transcriptions of references and data that do not contain advertisements (denoted as mono-extra-pub and multi-extra-pub).

Note that for both HMM models, the two distributions (intra-pub and extra-pub) for the Levenshtein distance are disjoint. This result means that by choosing an appropriate decision threshold for the Levenshtein distance, there is a big chance that all advertisements in radio streams can be detected.

As commonly found for speech recognition systems, at a phone-like level with current ALISP models the transcriptions of audio data are not perfect. Therefore, when two different repetitions of the same advertisement are analyzed there are differences (that is the reason why we need to apply the Levenshtein distance). The number of transcription errors is proportional to the length of the advertisement. For long advertisement, there is a greater risk to find more transcription errors that lead to bigger Levenshtein distance. On the other side, this study shows that as expected ALISP transcriptions made with multi-Gaussian model are more precise than those made with mono-Gaussian models.

4.3 Advertisement detection

To detect commercials in the test database we proceed as follows:

- Transcription of reference advertisements by 65 ALISP HMM models (acquired from the ALISP development data set).
- Transcription of the test data to obtain ALISP sequences for each radio.
- Setting the decision threshold on the development set of the Levenshtein distance to 0.75 for mono-Gaussian models and to 0.65 for multi-Gaussian models, to be sure to detect all advertisements.
- Searching for each ALISP transcription of the reference advertisement in the ALISP transcriptions of each test audio stream.

In order to evaluate the detection performance precision (P%) and recall (R%) rates are given in Table 2.

- Precision: The number of advertisements correctly detected / Total number of detected advertisements.
- Recall: The number of advertisements correctly detected / The number of advertisements that should be detected.

Note that with this system we want to detect known commercials in a radio stream. Moreover the system allows us to detect also some errors present in the manual annotations. Some errors in annotation files and some corrupted reference files were found as a result of these experiments.

Table 2 shows that for both HMM models, the system was not able to detect 72 advertisements. When analyzing the errors, we found two types of errors. The first one concerns commercials that are very short (3–5 s) and their references are shifted by 2 or 3 s. The second type of error is related to commercials which were different from the reference ones. For some of these missed advertisements, it was found that the manual annotations were incorrect.

By correcting the shifted reference files, our system was able to detect 2,021 commercials over 2,070. The 49 missed advertisements were different from the reference

Table 2 Precision (P%), recall value (R%), number of missed ads and number of false alarms found in each radio stream

Radio ID	P%		R%		Missed ads		False alarms	
	Exp1	Exp2	Exp1	Exp2	Exp1	Exp2	Exp1	Exp2
1	97	100	96	96	7	7	5	0
165038	92	100	95	95	4	4	7	0
3	96	99	96	96	8	8	8	2
541	88	100	97	97	2	2	9	0
547	95	100	96	96	6	6	8	0
548	96	100	96	96	10	10	7	0
552	94	100	97	97	5	5	11	0
553	94	99	98	98	3	3	10	1
555	91	100	97	97	2	2	8	0
557	98	100	95	95	12	12	3	0
558	98	99	95	95	13	13	4	2
Mean/*Total*	**94**	**99**	**96**	**96**	*72*	*72*	*80*	*5*

Results for the evaluation set (three days of audio stream, containing 2,070 commercials) with a threshold of 0.75 for mono-gaussian models (Exp1) and 0.65 for multi-gaussian models (Exp2). The bold data indicate mean values, while italics indicate number of advertisements

ones and there is no way to detect them with our system. These missed commercials could be divided into three groups. The first one concerns advertisements on TV or radio shows where the program changes daily or weekly, which involves changing the content of these commercials whenever the show is broadcasted. The second group is related to commercials pronounced by different speakers who say the same things. The last category is advertisements dealing with the new music albums release or music concert where the song broadcasted to promote the album or the concert is different from the one in the reference file.

We note the presence of 80 false alarms for the mono-Gaussian HMM models. The number of these false alarms was reduced to only five false alarms by using the multi-Gaussian HMM models. Actually, the multi-Gaussian system has confused some advertisements of the same products but with a slight change in their content and that were annotated differently by human annotators. For future experiments the ground truth annotations will be corrected for the shifted reference advertisements. Also a consensus for the naming of advertisements with varying content should be found.

Related to the processing time, three steps are considered. The acquisition and modeling of ALISP units is done off-line. When processing test data, the processing time needed to transcribe the audio streams with ALISP HMM models is negligible. The computational complexity of the system is currently limited by the search for the closest ALISP sequence with the Levenshtein distance. With the current implementation, to search for our reference advertisements (2,172 advertisements) in 1 h of radio stream, the time processing needed by the system is 1 h on a 3.00GHz Intel Core 2 Duo 4GB RAM.

5 Conclusions and perspectives

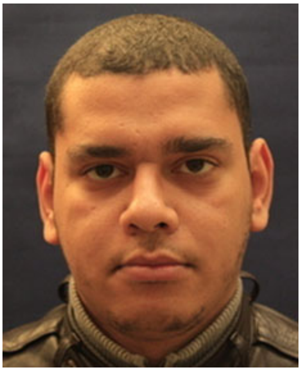
In this paper we present an indexing system to detect known advertisements in audio streams. This system is based on transcriptions provided by data-driven ALISP tools. The system is evaluated on one day broadcast corpus of 11 radios. 2,021 advertisements over 2,070 were correctly detected and some errors in the manual annotations were detected. Moreover, we show that the system based on the multi-Guassian HMM models is more efficient than the one based on mono-Gaussian HMM models in terms of false alarms. We are going to test this approach for music detection and compare it with existing music detection systems. Future work will be dedicated to improve the matching component by adapting the BLAST [2] algorithm which is often used to compare primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. We will also focus on detecting advertisements without references by extracting salient parts or by finding all repetitions of audio sequences in the entire database which should lead to the automatic discovery of advertisements, musical pieces and jingles.

Acknowledgements Some of the programs used for this research were written by F. Bimbot, J. Cernocky, A. El Hannani, G. Aversano and members of the French SYMPATEX project. The first author is financially supported by the ANR-SurfOnHertz project. We would also like to thank the reviewers for having accepted to revise our paper and for their precious advices and comments that have greadly improved the quality of the paper.

References

1. Allamanche E, Herre J, Hellmuth O, Froba B, Cremer M (2001) Audioid: towards content-based identification of audio material. In: 110th conv. of the AES
2. Altschul SF, Gish W, Miller W (1990) Basic local alignment search tool. J Mol Biol 215:403–410
3. Atal B (1983) Efficient coding of lpc parameters by temporal decomposition. In: ICASSP, pp 81–84
4. Bimbot F (1990) An evaluation of temporal decomposition. Tech. rep., Acoustic Research Department AT&T Bell Labs
5. Cambridge University Engineering Department. HTK: Hidden Markov Model ToolKit v 3.2.1. http://htk.eng.cam.ac.uk. Accessed Apr 2010
6. Cano P, Battla E, Mayer H, Neuschmied H (2002) Robust sound modeling for song detection in broadcast audio. In: Proc. 112th AES convention. Audio Engineering Society, Munich, Germany
7. Cano P, Battle E, Kalker T, Haitsma J (2005) A review of audio fingerprinting. J VLSI Signal Process 41(3):271–284
8. Cardinal P, Gupta V, Boulianne G (2010) Content-based advertisement detection. In: INTERSPEECH
9. Cernocký J (1998) Speech processing using automatically derived segmental units: applications to very low rate coding and speaker verification. PhD thesis, Université Paris XI, Orsay-France
10. Chollet G, Cernocký J, Constantinescu A, Deligne S, Bimbot F (1999) Towards ALISP: a proposal for Automatic Language Independent Speech Processing, NATO ASI Series. Springer, pp 357–358
11. Chollet G, McTait K, Petrovska-Delacrétaz D (2005) Data driven approaches to speech and language processing. Lecture Notes in Computer Science, pp 164–198
12. Covell M, Baluja S, Fink M (2006) Advertisement detection and replacement using acoustic and visual repetition. In: MMSP, pp 461–466
13. El Hannani A (2007) Text-independant speaker verification based on high-level information extracted with data-driven methods. PhD thesis, University of Fribourg (Switzerland) and INT/SITEVRY, France
14. El Hannani A, Petrovska-Delacrétaz D, Fauve B, Mayoue A, Mason J, Bonastre JF, Chollet G (2009) Text independent speaker verification. In: Guide to biometric reference systems and performance evaluation. Springer
15. Haitsma J, Kalker T (2002) A highly robust audio fingerprinting system. In: ISMIR

16. Jang D, Lee S, Lee JS, Jim M, Seo JS, Lee S, Yoo CD (2006) Automatic commercial monitoring for TV broadcasting using audio fingerprinting. In: AES 29th international conference
17. Levenshtein V (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10(8):707–710
18. Linde Y, Buzo A, Gray RM (1980) An algorithm for vector quantizer design. IEEE Trans Commun 28(1):84–95
19. Makhoul JM, Roucos S, Gish H (1985) Vector quantization in speech coding. Proc IEEE 73(11):1551–1588
20. Neves C, Veiga A, Sá L, Perdigo F (2009) Audio fingerprinting system for broadcast streams. Proc Conf Telecommunications 1:481–484
21. Padellini M, Capman F, Baudoin G (2004) Very low bit rate (vlbr) speech coding around 500 bits/s. In: EUSIPCO
22. Pinquier J, Andre-Obrecht R (2004) Jingle detection and identification in audio documents. ICASSP 4:329–322
23. Schwarz D (2004) Data-driven concatenative sound synthesis. Acoustics, Computer Science, Signal Processing Applied to Music, Université Paris 6—Pierre et Marie Curie
24. Wang A (2006) The shazam music recognition service. Commun ACM 49(8):44–48
25. Young S, Russell N, Thornton J (1989) Token passing: a conceptual model for connected speech recognition systems. Tech. rep., Cambridge University



Houssemeddine Khemiri was awarded an engineering degree in telecommunication from the Tunisian High School of Telecommunications (Sup'Com) in 2008. In 2009, he was granted a Master degree in Multimedia Applications from the Tunisian High School of Telecommunications. Since 2010, he is PhD student in Telecom ParisTech under the supervision of Gérard Chollet and Dijana Petrovska-Delacrétaz, working on audio indexing with data-driven approaches.

Gérard Chollet was granted a PhD in Computer Science and Linguistics from the University of California, Santa Barbara. He joined CNRS (the French public research agency) in 1978. In 1981, he took in charge the speech research group of Alcatel. In 1992, he participated to the development of IDIAP, a new research laboratory of the 'Fondation Dalle Molle' in Martigny, Switzerland. Since 1996, he is full time at ENST, managing research projects and supervising doctoral work. His main research interests are in phonetics, automatic audio-visual speech processing, speech dialog systems, multimedia, pattern recognition, biometrics, digital signal processing, etc.

Dijana Petrovska-Delacrétaz obtained her degree in Physics and PhD from the Swiss Federal Institute of Technology (EPFL) in Lausanne. She was working as a Consultant at AT&T Speech Research Laboratories, as a post-Doc at Télécom ParisTech, and as a Senior Scientist at the Informatics Department of Fribourg University, Switzerland. She currently holds an associate professor position within Télécom SudParis. Her research activities are oriented towards pattern recognition, signal processing, and data-driven machine learning methods that are exploited for different applications such as speech, speaker and language recognition, very low-bit speech compression, biometrics (2D and 3D face, and voice), and crypto-biometrics (including privacy preserving biometrics). As per July 2011, her full list of publications (see also http://webspace.it-sudparis.eu/~petrovs) is composed of: three patents, two publicly available databases (for speaker recognition and biometrics evaluations), one collection of open-source software, one book, 17 book chapters and journal papers, and 50 publications in conferences proceedings.