INTERSPEECH 2009 BRIGHTON

Audio keyword extraction by unsupervised word discovery

Armando Muscariello, Guillaume Gravier, Frédéric Bimbot

IRISA - Metiss Research Group, France
CNRS - UMR 6074
INRIA - Rennes Bretagne Atlantique
name.surname@irisa.fr

Abstract

In real audio data, frequently occurring patterns often convey relevant information on the overall content of the data. The possibility to extract meaningful portions of the main content by identifying such key patterns, can be exploited for providing audio summaries and speeding up the access to relevant parts of the data. We refer to these patterns as audio motifs in analogy with the nomenclature in its counterpart task in biology. We describe a framework for the discovery of audio motifs in streams in an unsupervised fashion, as no acoustic or linguistic models are used. We define the fundamental problem by decomposing the overall task into elementary subtasks; then we propose a solution that combines a one-pass strategy that exploits the local repetitiveness of motifs and a dynamic programming technique to detect repetitions in audio streams.
Results of an experiment on a radio broadcast show are shown to illustrate the effectiveness of the technique in providing audio summaries of real data.
Index Terms: unsupervised learning, motif discovery, audio summary, dynamic programming.

1. Introduction

Audio compression and digital storage techniques have made it increasingly possible to store large amount of audio data on cheap electronic devices. This has in turn implied the need for smart strategies to quickly browse the data and summarize the audio, as studied as part of the Quaero project[1]. The identification of repeating patterns in the data addresses these two issues: assuming that repeating excerpts carry meaningful information on the main content of the data, such patterns can be directly used to summarize the audio; besides, by knowing their locations in the stream, a user is enabled to rapidly access relevant audio segments. The process can be regarded as the audio counterpart of the key frame extraction task in video: the salient structure of the video sequence is grasped by a small number of representative still images; similarly, we aim at providing an acoustic summary of an audio recording by means of keywords in the audio form.
In our framework, the discovery is performed in an unsupervised fashion; the a priori knowledge on the targeted patterns is minimal: properties like motif length, number of occurrences, location in the stream, amount of variability allowed among motif occurrences, are not known or hypothesized prior to the processing step. We only constraint the mimimum admissible motif length to avoid the risk of matches as short as a few samples of signal. Besides, the process is merely unimodal, meaning that no additional clues about the presence of motifs is gathered from video or textual sources. This approach notably differs from the more traditional *train and test* strategies common to most supervised models for recognition, as in modern ASR systems, where complex acoustic and linguistic models are designed and trained on large amount of data prior to the discovery. We also insist in highlighting the absence of language models in our scenario, while acoustic models are learnt directly during the actual discovery and refined as more input is processed. With respect to ASR solutions, this approach presents several advantages: it is language independent, it does not suffer of the out of vocabulary problem and is sensibly faster than alternative approaches that rely on a textual transcription of the audio.
In recent years, a few works have formalized and addressed the problem of audio motif discovery. In [1] repetitions of jingles and songs in multimedia streams are detected by time correlating low-dimension audio fingerprints in a framework that exploits statistical properties of real streams to speed up the discovery. [3] and [2] have in common a two pass strategy to discover word and word-like fragments in speech data: the first pass to retrieve similar segments by novel dynamic programming techniques and the second one to group them into homogeneous clusters, each representing a motif. Our approach aims at combining the one pass strategy in [1] suitable for on-line processing of streams and the similarity detection techniques used in [2] and [3], suitable for information retrieval experiments on speech data. While similar to our work in [4], the solution proposed here overcomes some of the limitations of the previous approach, by relaxing the constraints on the motif location in the stream and by freeing the algorithm from the necessity to fix the motif maximum length.
The paper is organized as follows: in section 2 the problem is formalized and decomposed into subtasks; each of them is considered in each subsection of section 3; finally experiments on real data are discussed in section 4.

2. Formalization of the problem

Motif discovery in a stream χ can be seen as the problem of finding all the pairs $[a, b]$ and $[c, d]$ in χ subject to:

$$H(\chi_a^b, \chi_c^d) < \epsilon \quad (1)$$

$$|b - a| > L_{min} \quad (2)$$

$$a < b < c < d \quad (3)$$

Eq. (1) states that two segments χ_a^b and χ_c^d are occurrences of a motif if the cost function $H(\cdot)$ applied to the pair χ_a^b, χ_c^d is below a given threshold ϵ. Condition (2) imposes a constraint on the motif minimum length while (3) prevents considering over-

[1] http://quaero.org

lapping segments as instances of the same motif to avoid trivial matches.
This formulation aims at discovering motifs by detecting similarities in audio in a pairwise manner and grouping two occurrences of a motif into a single cluster. Different clusters containing occurrences of the same underlying motif must be merged afterwards in order to retain only one representative cluster of all the occurrences of a motif in the stream. This consideration suggests to view at motif discovery as a clustering technique that operates only on the part of the data that effectively repeats. This formulation applies both when the data is available all at once in a file and processed in a two-pass strategy (the first one to discover similarities, the second one to form clusters of motifs), and in *streaming mode*, where the data is progressively received and clusters are incrementally built in just one pass over the data.
Taking into account these considerations, we propose to decompose the overall discovery task into four elementary subtasks:

1. Organization of the motif discovery process: the structural way to segment the data and organize the search for repetitions (this includes the choice between a one-pass or two-pass strategy).
2. Feature extraction: the extraction of appropriate features from the signal to obtain a representation well suited for the similarity detection task.
3. Similarity detection: the identification in χ of the couples χ_a^b and χ_c^d as the most likely segments to be occurrences of a motif.
4. Similarity score: the computation of a distance measure to deem whether two audio segments are similar or not. This is equivalent to defining and computing the cost function H in (1) applied to the couples χ_a^b and χ_c^d, and verifying their similarity against the threshold ϵ.

Note that the last two subtasks, though conceptually different, can be integrated into the same one, depending on the solution proposed. The couples χ_a^b and χ_c^d can be identified as the segments that minimize H and the same score can be used as a similarity score. This is the solution that we have adopted. However, tasks 3 and 4 can be distinct if similarity detection is based on some kind of approximate fast pattern matching technique. Each of these tasks will be tackled in detail in the following.

3. Algorithm

3.1. Organization of the motif discovery

The proposed solution works by incrementally building and updating a library of motifs that are discovered as the incoming stream is received. The local stream is decomposed into two adjacent audio segments called the "seed block" and the "near future". The seed block is considered as a potential fragment of a motif which is to be searched a) as a sub-segment in the library of already existing motifs and, if not found there, b) as a possible match in the "near future". If any of these two situations occur, a full search is triggered by extending the seed block leading to one of the following three possibilities : the detection of a new utterance of an already existing motif, the discovery of a new motif or the discarding of the current seed block and the shift to the next seed block. This one pass approach, depicted in Figure 1 is conceptually identical to ARGOS [1].The underlying assumption is that, in real streams, frequently occurring patterns, like topic-specific terms, are likely to repeat in a relatively short time span. Thus by exploiting this

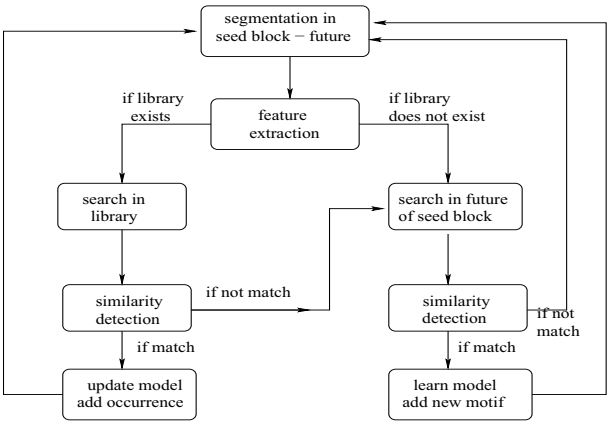

Figure 1: The diagram refers to the single step of the iterative process. At each step the seed block is compared with the models in the library and, if not found, is searched in its near future. At the following step the pair 'seed block-future' shifts of a seed block length along the audio stream and the process is iterated.

local repetitiveness, short term occurrences are first discovered in the future of the seed block, and the long term ones by library search, avoiding brutal force strategies that require segments of all admissible lengths to be assumed as candidate motifs and searched over the entire stream.
Let us consider in the stream χ a time window $[t_0 - \Delta, t_0]$ with $\Delta = \frac{L_{\text{min}}}{2}$ where L_{min} is the motif minimum length and its near future $[t_0, t_0 + L_{\text{fut}}]$. We call $\chi_{t_0-\Delta}^{t_0}$ the seed block.
If there exists a segment χ_c^d with $t_0 < c < d < t_0 + L_{\text{fut}}$ such that $H(\chi_{t_0-\Delta}^{t_0}, \chi_c^d) < \epsilon$ we define the seed block as the *seed match*. The seed match is not a motif as its length is half the motif minimum length. However, we can check the existence of a match by extending the seed match, using $[t_0 - \Delta, t_0]$ and $[c, d]$ as anchor points. If there exist the couples $a' < t_0 - \Delta, b' > t_0$ and $c' < c, d' > d$ such that:

$$H(\chi_{a'}^{b'}, \chi_{c'}^{d'}) < \epsilon \tag{4}$$

$$H(\chi_{a''}^{b''}, \chi_{c''}^{d''}) > \epsilon,\ \forall a'' < a', b'' > b', c'' < c', d'' > d' \tag{5}$$

$$|b' - a'| \geq L_{\text{min}} \tag{6}$$

then a match is found with occurrences $\chi_{a'}^{b'}$ and $\chi_{c'}^{d'}$.
The condition in (5) signifies that the match can not be further extended outside $[a', b']$ and $[c', d']$ without infringing the similarity condition in (1). From these two occurrences of a motif, a model is learnt and stored in a library as a reference model for the retrieval of long term occurrences.
At the next step, a new seed block $\chi_{b'}^{b'+\Delta}$ and its near future are considered. First the seed block is searched in the motifs already in the library and, if not found, in its near future. If $\chi_{t_0-\Delta}^{t_0}$ was not a seed match, then the next seed block is $\chi_{t_0}^{t_0+\Delta}$ and its near future $\chi_{t_0+\Delta}^{t_0+\Delta+L_{\text{fut}}}$.
The reason for choosing $L_{\text{min}}/2$ long seed blocks is straightforward: since a motif needs to be at least L_{min} long, it is guaranteed that a motif, whatever its position in the stream, completely includes at least one of the $L_{\text{min}}/2$ seed blocks the stream is divided into. A similar one pass approach was already adopted in our previous work [4]. In [4] the goal was to discover a matching subsequence between the motif candidate, called the query, and its near future, called the search buffer; here the goal is to

discover a repetition of the whole seed block in its future and then extend the matching part to retrieve the whole match. The approach introduced in this work differs from the previous one by at least two main factors: a) in the current version there is no need to set a motif maximum length that coincides with the query length, as in [4], and b) in [4] matches might not be detected if a part of the motif exceeds the query boundaries, as matches can not be extended outside the query boundaries.

3.2. Audio signal representation

In [1] the audio signal is fingerprinted by only retaining a small portion of the power spectrum (about 200 Hz centered around the sixth Bark band): good results are observed when applied to the detection of repetitions of jingles and songs in broadcast radio shows. However, speech signal exhibits an intrinsic variability that makes this technique for audio content fingerprinting unsuitable for discovering repeating words. Accordingly, as in [2] and [3], we resort to the classic representation of the audio signal by MFCC coefficients. We extract one energy coefficient and 12 dimensional MFCC every 10 ms.

3.3. Audio similarity detection

In the current version of the algorithm, the similarity detection and similarity score subtasks are solved in a unique procedure. We use the same technique for both identifying the segment most likely to be a repetition of the seed block in its future, and to compute the similarity score between these two *motif occurrences' candidates*. We resort to the first technique we introduced in [4], called *segmental locally normalized dynamic time warping* (SLNDTW). It is a dynamic programming technique that aims at detecting low distortion local alignments between the seed block and its future, by operating in a three-stage procedure: the first one to identify starting points of potential matching paths in the distance matrix; the second one to compute the best path; the third one to extend the matching path (if any) outside the boundaries of the seed block to retrieve the entire occurrence.
We compute the MFCC vectors of the seed block and its future, respectively $\{u_i\}_{i=1}^{M}$ and $\{v_j\}_{j=1}^{N}$, and their Euclidean distance matrix $d(i,j), 1 \leq i \leq M, 1 \leq j \leq N$. We define $L(i,j)$ as the length of the path starting from some $(1,j)$ up to (i,j), $D(i,j)$ the corresponding accumulated distance (the sum of the local distances for each entry of the path) and $W(i,j) = D(i,j)/L(i,j)$ its average weight.

3.3.1. Identification of the starting points

While scanning the first row of the distance matrix, the algorithm selects an entry (i,j) as the starting point of a new path if its local distance $d(i,j)$ is *low enough* compared to the weight of the path passing through its left neighbour; indeed the low value of $d(i,j)$ could signal a matching path starting there. Formally: $\forall j, 1 \leq j \leq N$,

$$\begin{cases} D(1,j) = d(1,j) \\ L(1,j) = 1 \end{cases} \quad \text{if } d(1,j) < W(1,j)$$

$$\begin{cases} D(1,j) = D(1,j-1) + d(1,j) \\ L(1,j) = L(1,j-1) + 1 \end{cases} \quad \text{otherwise}$$

3.3.2. Path computation

Except for $i = 1$, each path is computed by iteratively applying the dynamic programming relations following the local normalization paradigm, which consists in minimizing, at each point (i,j) the weight $W(i,j)$, that is the quotient between the accumulated distance $D(i,j)$ and the path length $L(i,j)$. Formally:

$$W(i,j) = \min \left[\frac{d(i,j) + D(i-1,j)}{L(i-1,j)+1}, \frac{d(i,j) + D(i-1,j-1)}{L(i-1,j-1)+1}, \frac{d(i,j) + D(i,j-1)}{L(i,j-1)+1} \right] \quad (7)$$

The ending point (M, j_e) of a match is such that $W(M, j_e) < \phi, 1 \leq j_e \leq N$, where ϕ is a given spectral threshold. The average path weight W and the spectral threshold ϕ play the role respectively of the cost function $H(\cdot)$ and the threshold ϵ defined in (1). The path is then retrieved by applying the dynamic programming relations in reverse order starting from the ending point of the match (backtracking).

3.3.3. Path extension

The extension of the match is achieved by extending the matching path. The boundaries $(1, j_s)$ and (M, j_e) of the matching path are used as anchor points, to which new frames are appended as long as the average weigth W of the extended path does not exceed the defined threshold (that is, as long as the similarity condition in (1) is satisfied). For example, while extending the path from the ending point, the local distances of the neighbourhood of (M, j_e) are computed (the frames $(M, j_e + 1)$, $(M + 1, j_e + 1)$ and $(M + 1, j_e)$); then the minimization in (7) is carried out to choose the best new frame to add, and its contribution to the average weight of the path is evaluated according to (1) to decide whether to further extend the path or to stop. Figure 2 illustrates the scenario.
If several repetitions of the seed block occur in its future, we only retain the first one; the other ones will be detected in the following iterations of the algorithm by comparison with the corresponding model stored in the library.

3.4. Motif model and library search

Once two occurrences of a motif are detected in the seed block-future comparison, a model of the motif is learnt and stored in a library. In our current implementation the model is built by simply averaging the spectral frames put in correspondence by the matching path computed by SLNDTW and path extension. Once a library of the motif is initialized, the new seed blocks are compared first with those models. The comparisons are carried out by SLNDTW and match extension as in the seed block-future scenario already described; if a match is detected, the model is updated.

4. Experiments and results

The algorithm has been tested on a one hour long French radio broadcast show, sampled at 16 kHz. The evaluation of the performance of the algorithm is carried out at the phonetic level by relying on phonetically aligned version of the data. For each motif found, the phonetic sequences of the occurrences, are searched into the phonetic alignment, and compared by normalized edit distances [5]. A centroid of the motif is identified as the sequence closest to all the other ones according to the edit distances computed. The precision is the fraction of them close to the centroid, *i.e.* for which the distance to the centroid lies within a specified value θ. Afterwards, all the sequences

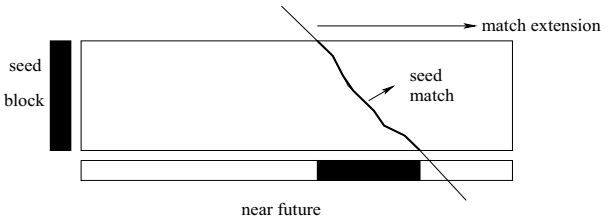

Figure 2: SLNDTW: after a seed match is found, the matching path is extended to retrieve the whole match by using the same relation in (7) used to compute each path.

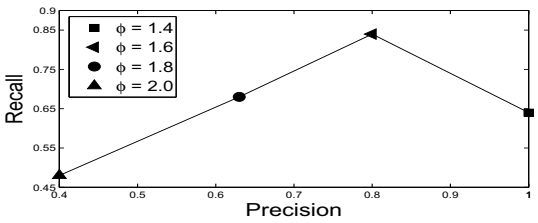

Figure 3: Precision and recall measures for various values of ϕ.

in the phonetic alignment that are close to the centroid are retrieved: the recall is the fraction of them already discovered by the algorithm at the acoustic level. A rigorous definition of these quantities can be found in [4].

As for the main parameters, we have set $L_{\text{min}} = 0.6$ seconds since the frequently repeating patterns are mostly words or small multi-words phrases; we have fixed $L_{\text{fut}} = 60$ seconds as the duration of each news report, where most of the topic specific terms are expected to occur, is around one minute on average. We have used unitary weights for the elementary edit operations in the computation of the edit distances. Therefore the normalized edit distance defined in [5] ranges from zero to one. We have chosen $\theta = 0.35$ to discriminate the similarity between two phonetic sequences; this is because, even in the case of correct discovery, the occurrences found usually differ by some phonemes at the boundary since, in the match extension stage, high distortions frames are added to the matching path as long as the similarity condition in (1) is satisfied.

The algorithm is run for different values of the spectral threshold ϕ and the corresponding precision-recall pairs are reported in Figure 3: as expected, precision increases as ϕ becomes lower, because the number of false hits is reduced as the similarity condition becomes more selective. At the lowest thresholds, precision values (close to one) show a significant level of purity achieved by the algorithm. As for the recall, a transition can be observed at $\phi = 1.6$: recall increases from $\phi = 1.4$ to $\phi = 1.6$ then progressively decreases at higher thresholds. This is only apparently counterintuitive, as one could expect higher recall and lower precision values as the threshold increases, since more false and true hits are detected. However, averaging false hits with the model in the library tends to progressively reduce its representativeness, leading to missed detection of true instances of the motif afterwards.

From a qualitative point of view, the algorithm was able to discover several words useful to characterize the context of each news, which is indeed the aim of an audio summary: for example the terms *Declaration des droits de l'homme* and *prix Nobel de la paix* were discovered inside the audio segment concerning the celebration of the 50th anniversary of the Universal Declaration of Human Rights attended by several Nobel Peace Prize Laureates; the term *controlleurs* when talking about the union demands of French train officials, or the names of various characters *Jean Castellat*, *Francis Lamarque*, *Jeannie Longo*, useful to rapidly identify the topic of a given news.

Different kind of shortcomings or errors can be deduced by observing some of the segments detected. The short breathings in between words and silence intervals are example of repetitions that do not add useful information for the audio summary, that the algorithm identifies nonetheless. In the current version, besides, the algorithm is strongly speaker dependent: when the same word is uttered several times by different speakers, the algorithm usually collect in the same cluster all the occurrences sharing the same speaker. Another type of shortcoming can be observed, especially for high values of ϕ, when different words are deemed as similar when only sharing a subword (for example, in one case, the words *discussion* and *illusion* are recognized as occurrences of a same motif, although having in common only the ending part). As far as the computation time is concerned, a C implementation of the algorithm (not thouroughly optimized) is able to process the entire one hour long stream in ten minutes for the lowest value of ϕ and in almost forty minutes for the highest one. Indeed for increasingly high thresholds the number of motifs discovered increases in turn, and more time is needed to scan the library and compare the seed block with the models stored.

5. Conclusions and future works

In this paper, a framework has been proposed for the discovery of repeating words in audio streams without relying on predefined acoustic and linguistic models, with the aim of producing audio summaries by keyword extraction. The algorithm combines a one pass strategy for online processing of streams that exploits the local repetitiveness of motifs and a dynamic programming based technique for the detection of similarities in audio.

Although good preliminary results have been observed, several deficiencies of the current implementation need to be addressed in the future: a better modeling of the motif, for example by using probabilistic models, is needed to improve the similarity detection and prevent the rapid degradation of the model for high values of spectral thresholds. Moreover, speaker normalization techniques will also be considered to account for situations where utterances underlying the same lexical pattern are spoken by different speakers. Finally, research will focus on making the time computation independent from the size of the library, by either adopting fast approximate pattern matching techniques or by limiting the comparisons to a constant number of promising models in the library.

6. References

[1] Herley, C.: ARGOS: Automatically Extracting Repeating Objects from Multimedia Streams. IEEE Transactions on Multimedia, VOL. 8 (2006).

[2] Park, A., Glass, J.R.: Unsupervised pattern discovery in speech: IEEE Transaction on Acoustic, Speech and Language Processing, VOL. 16, (2008).

[3] Ten Bosch, L., Cranen, B.: A Computational model for unsupervised word discovery. Interspeech 2007, pg: 1481-1484.

[4] Muscariello, A., Gravier, G., Bimbot, F.: Variability tolerant audio motif discovery. Multimedia Modeling 2009.

[5] Vidal, E., Marzal, A., Aibar P.: Fast Computation of Normalized Edit Distances. IEEE Transaction on Pattern Analysis and Machine Intelligence, VOL 17, NO. 9, (1995).