Variability tolerant discovery of arbitrary repeating patterns in audio data

Armando Muscariello

N° d'ordre : 4322 ANNÉE 2011

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale Matisse

présentée par

Armando Muscariello

préparée à l'IRISA

Institut de Recherche en Informatique et Système Aléatoires
Composante universitaire: IFSIC

Variability tolerant discovery of arbitrary repeating patterns in audio data with template matching

Thèse soutenue à Rennes le 25 Janvier 2011

devant le jury composé de :

Gérard CHOLLET
Directeur de Recherche CNRS à Telecom ParisTech, Paris / président

Régine ANDRÉ-OBRECHT
Professeur à l'IRIT, Toulouse / rapporteur

Jan ČERNOCKÝ
professeur à Brno University of Technology, Brno / rapporteur

Xavier ANGUERA MIRÓ
chercheur à Telefonica I+D, Barcelona / examinateur

Raphaël BLOUET
ingenieur à Yacast, Paris / examinateur

Guillaume GRAVIER
chargé de recherche CNRS à l'IRISA, Rennes / directeur de thèse

Frédéric BIMBOT
Directeur de Recherche CNRS à l'IRISA, Rennes / directeur de thèse

Contents

List of Figures

Chapter 1

Resumé en français

1.1 Motivation

Dans de nombreuses applications, il est utile de résumer un contenu afin d'en permettre une apprehension rapide. Ainsi, pour les textes, on a généralement recours à quelques mots ou phrases clés tandis qu'en vidéo, on utilise des images clés présentées sous forme d'icônes. En revanche, appréhender un contenu audio directement à partir du signal reste problématique. Dans le cas de contenus oraux, il est évidemment possible d'utiliser une transcription automatique pour se ramener au cas du texte. Mais le processus de transcription automatique est coéteux et parfois peu fiable. La détection de mots clés, ou *word spotting*, présente une alternative intéressante mais limitée à une liste de mots prédéfinis.

Nous étudions ici une approche radicalement différente basée sur la découverte de motifs dans le signal pour faire émerger des icônes sonores correspondant à des mots ou des locutions caractéristiques d'un contenu. La découverte de motifs sonores consiste à détecter à partir du signal des éléments acoustiques récurrents présentant éventuellement un certain degré de variabilité, sans aucune forme de connaissance *a priori*, tant sur le plan acoustique que linguistique. Par exemple, dans le cas de la parole, les mots ou locutions qui se répétent sont des motifs typiques que nous souhaitons voir émerger.

Il convient de bien distinguer la *découverte* de motifs de la *recherche* de motifs. Dans le premier cas, les motifs ne sont pas définis *a priori* tandis que dans le deuxième cas, il s'agira de retrouver un motif connu et défini à l'avance, par exemple par une occurence de référence. Par ailleurs, il est également important de noter que nous

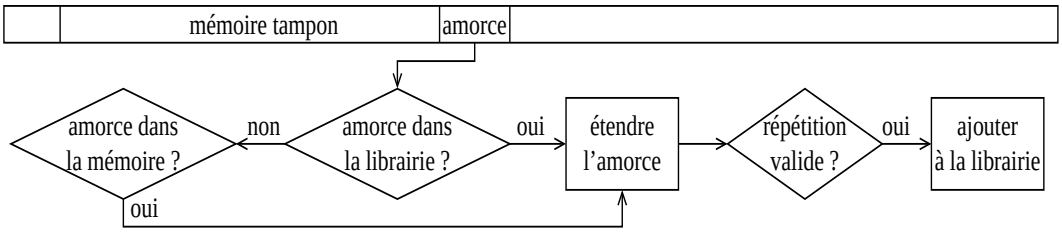

Figure 1.1: Schéma de principe de la segmentation du flux et de la recherche pour une amorce donnée.

souhaitons développer des approches non supervisées dans lesquelles aucune forme d'apprentissage n'intervient. En particulier, nous ne souhaitons utiliser ni modéle de langage, ni modéle acoustique prédéfinis.

Dans le domaine audio, quelques travaux récents s'intéressent au problème de la découverte de motifs. En particulier, Herley propose un algorithme de découverte de motifs sonores quasi invariants pour la découverte d'éléments récurrents (génériques, publicités, *etc.*) dans un flux télévisé Herley (2006). De récents travaux sur la découverte de mots dans le signal de parole relève le défi de la variabilité des motifs Muscariello *et al.* (2009b); Park & Glass (2008); ten Bosch & Cranen (2007). Les approches proposées dans ten Bosch & Cranen (2007) et Park & Glass (2008) s'appuient sur un algorithme en deux passes : une première passe vise à détecter des fragments similaires qui sont regroupés dans une passe suivante. Dans Muscariello *et al.* (2009a), nous proposons une approche combinant la stratégie en une passe de Herley (2006) avec les méthodes de comparaison de séquences basées sur l'alignement temporel dynamique (DTW). Dans cet article, nous étendons l'approche présentée dans Muscariello *et al.* (2009a) afin d'accroître la robustesse de l'algorithme à la grande variabilité du signal de parole.

Nous formalisons tout d'abord le problème de la découverte de motif avant de détailler l'architecture générale de l'approche proposée. Nous détaillons à la section 1.4 différentes méthodes pour la comparaison de deux séquences sonores. Les résultats expérimentaux sont rassemblés dans la section 1.5.

1.2 Formalisation du problème

De manière tout à fait générique, la découverte de motifs consiste à trouver dans un ensemble de données ϕ toutes les paires de segments disjointes, de longueur minimal $L_{\min}$, suffisamment proches. Formellement, on cherche les paires ϕ_a^b, ϕ_c^d telles que

$$H(\phi_a^b, \phi_c^d) < \epsilon \ , \tag{1.1}$$

où H est une mesure de la distance entre les deux segments, sous les contraintes $b - a > L_{\min}$ et $a < b < c < d$.

Ainsi formulée, la découverte de motifs a pour but de trouver des paires de segments similaires, regroupant ainsi deux occurences d'un même motif. Une étape supplémentaire de *clustering* est ensuite nécessaire pour grouper l'ensemble des occurences d'un motif. Une telle considération nous amène è envisager le problème de découverte de motifs comme un problème de *clustering* se limitant aux portions de signal qui se répètent au moins une fois. Une telle approche s'applique aussi bien lors d'un traitement *a posteriori*, par exemple avec une stratégie multipasse lorsque l'ensemble des données est accessible ten Bosch & Cranen (2007); **?**, que pour un traitement en flux Herley (2006); Muscariello *et al.* (2009a)

Du point de vue conceptuel, nous pouvons décomposer la découverte de motifs en quatre tâches élémentaires : représentation, segmentation, détection et validation. La *représentation* consiste à choisir les descripteurs utilisés pour représenter le signal. La *segmentation* recouvre l'organisation du processus en terme de segmentation des données et d'organisation de la recherche. En effet, une recherche exhaustive de toutes les paires vérifiant (1.1) n'est bien évidemment pas possible et le recours à une forme de segmentation s'avère indispensable. En particulier, le premier choix à effectuer est celui de la stratégie en une ou plusieurs passes. Enfin, les deux dernières tâches sont directement liées à la comparaison de segments et à la découverte des motifs. La *détection* consiste à identifier les répétitions ϕ_a^b, ϕ_c^d susceptibles de correspondre à deux occurences d'un motif. La *validation* permet par la suite de décider si deux répétitions correspondent en effet à un motif. Cette dernière tâche revient à vérifier (1.1). Bien que conceptuellement différentes, les tâches de détection et de validation peuvent se résumer en une seule si la même métrique H est utilisée pour les deux.

1.3 Architecture générale

Nous proposons une approche permettant un traitement en flux des données, dérivée de l'approche ARGOS Herley (2006) pour la segmentation. L'idée générale consiste à construire séquentiellement, de manière incrémentale, un catalogue de motifs à partir des données vues comme un flux. Dès lors qu'une nouvelle répétition est trouvée et validée, une nouvelle entrée est créée dans le catalogue, permettant ainsi de retrouver ultérieurement d'autres occurences de ce motif.

La détection des répétitions exploite la notion d'amorce, une amorce correspondant à un segment court, de taille fixé, dans le flux. Une amorce est vue comme un fragment de motif potentiel dont on cherche, dans la phase de détection, à trouver une répétition. Si une répétition de l'amorce est trouvée, on étend alors les segments répétés pour déterminer la répétition la plus longue possible. Cette répétition est ensuite validée comme occurence d'un motif dés lors que les deux segments sont suffisamment proches et insérée dans le catalogue. Afin de limiter le coét calculatoire et de permettre un traitement en flux, la recherche d'une répétition d'une amorce $\phi_t^{t+\delta}$ est limité au passé immédiat $\phi_{t-\Delta}^{t}$ conservé dans une mémoire tampon. La taille de l'amorce est étroitement liée à la taille minimum des motifs. En effet, l'amorce correspond à un hypothétique fragment de motif et, dans la mesure oé l'on cherche une répétition de l'amorce compléte, il est important qu'elle ne contienne pas de signal n'appartenant pas au motif lorsque l'amorce est effectivement un fragment de motif. Pour garantir cette propriété, on fixe $\delta = L_{\min}/2$.

Les étapes de l'algorithme sont illustrées par la figure **??**. Pour une amorce donnée $\phi_t^{t+\delta}$, on cherche dans un premier temps si cette amorce fait parti d'un motif connu, référencé dans le catalogue, ce dernier étant initialement vide. Si oui, on étend alors l'amorce pour vérifier qu'elle correspond au motif référencé dans le catalogue, remettant à jour le modéle du motif dans le catalogue le cas échéant. Dans nos travaux, le modèle de chaque motif est obtenu par moyennage des occurences trouvées. Si aucun motif du catalogue ne correspond, on cherche dans la mémoire tampon si il existe une répétition de l'amorce de manière à trouver deux occurences candidates pour un nouveau motif par extension de l'amorce. Si un nouveau motif est ainsi découvert, il est ajouté au catalogue après validation. L'algorithme se poursuit ensuite à partir d'une nouvelle amorce localisée soit juste après l'amorce courante si aucun motif n'a été trouvé, soit juste après l'occurence de motif trouvé.

1.4 Détection et validation

Dans le cadre de segmentation que nous venons de présenter, les tâches de détection et de validation interviennent à deux niveaux, lors de la comparaison avec les entrées du catalogue et lors de la recherche d'une répétition dans la mémoire tampon. Nous décrivons tout d'abord une technique de détection de motifs candidats utilisant une variante segmentale de la technique d'alignement temporel dynamique (DTW) avant de discuter de la validation des répétitions comme occurences d'un motif.

1.4.1 Détection par DTW segmentale

Rappelons tout d'abord que la phase de détection d'une répétition é partir d'une amorce est un processus en deux étapes. On cherche une répétition de l'amorce – dans le catalogue ou dans la mémoire tampon – avant d'étendre la correspondance de manière à trouver le fragment répété le plus long possible. Nous rappelons ici le principe général de ces deux étapes décrites en détail dans Muscariello *et al.* (2009a).

Considérons une amorce $\phi_t^{t+\delta}$ à rechercher dans un segment χ de longueur $l \gg \delta$. Cette recherche se fait par un algorithme de DTW dans lequel les contraintes de début et fin d'appariement sont relachées, de manière à trouver le fragment de χ apparié au mieux avec l'amorce. Le résultat est un segment χ_s^e tel que sa distance à l'amorce, normalisée par la longueur du chemin d'appariement, notée $D_{\text{DTW}}(\phi_t^{t+\delta}, \chi_s^e)$, est minimum. Les deux segments sont considérés comme une répétition si $D_{\text{DTW}}(\phi_t^{t+\delta}, \chi_s^e) < \epsilon_1$.

La deuxième étape vise à étendre au maximum à gauche et à droite l'appariement existant en s'appuyant sur les points extrêmes. Si l'on prend pour exemple le cas de l'extension à droite (*i.e.*, vers le futur) à partir des deux points $(\chi_e, \phi_{t+\delta})$, on cherche par DTW la meilleure extension vers $(\chi_{e+1}, \phi_{t+\delta+1})$, $(\chi_{e+1}, \phi_{t+\delta})$ et $(\chi_e, \phi_{t+\delta+1})$. Le processus d'extension se poursuit tant que D_{DTW} le long du nouvel appariement est inférieure é ϵ_1. Le résultat est une paire de segments, $\phi_{t-\beta_a}^{t+\delta+\alpha_a}$ et $\chi_{s-\beta_b}^{e+\alpha_b}$ telle que $D_{\text{DTW}}(\phi_{t-\beta_a}^{t+\delta+\alpha_a}, \chi_{s-\beta_b}^{e+\alpha_b}) < \epsilon_1$, correspondant à une hypothése de motif qu'il convient de valider.

L'étape de validation consiste à évaluer (1.1). La distance D_{DTW} peut étre directement utilisée comme métrique H. Cependant, afin d'éviter de valider deux segments différents, cette stratégie requiert un seuil ϵ_1 très petit, limitant ainsi la variabilité tolérée entre occurences d'un motif. En particulier, nous avons observé que cette

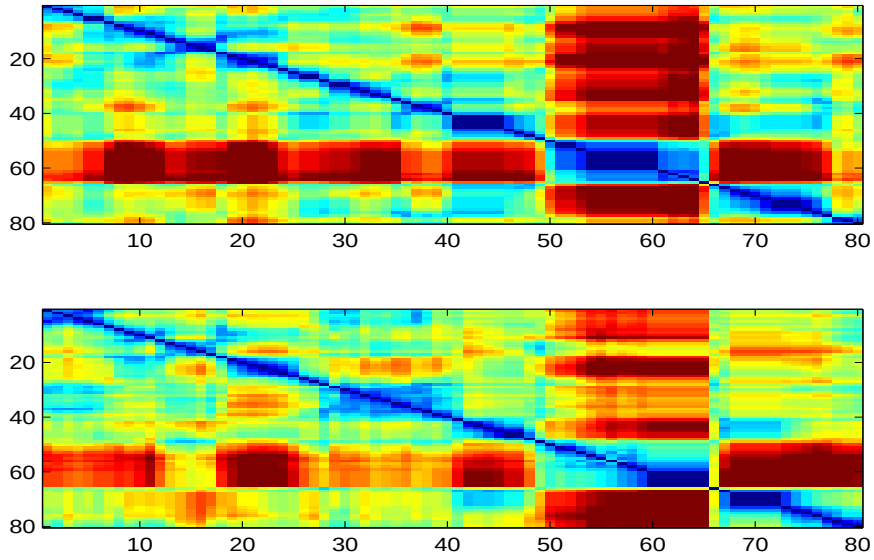

Figure 1.2: Exemple de matrices d'autosimilarité d'un motif pour deux locuteurs (masculin en haut, féminin en bas).

approche ne permet pas de retrouver des occurences d'un motif par différents locuteurs. Utiliser un seuil ϵ_1 plus élevé autorise une plus grande variabilité au prix d'un nombre plus élevé de fausses détections, c'est-é-dire de détection de répétitions ne correspondant pas à deux occurences d'un motif.

1.4.2 Validation par matrices d'autosimilarité

Pour pallier au problème précédent, nous proposons une étape de validation exploitant la comparaison de matrices d'autosimilarité. La matrice d'autosimilarité d'une séquence χ_a^b est la matrice carrée $\Phi(\chi_a^b)$ des distances entre points χ_i et χ_j. Clairement, les matrices d'autosimilarité de différentes occurences d'un motif présentent une forte ressemblance visuelle comme illustré par la figure **??**. C'est cette ressemblance – interprétable comme une distance entre les autocorrélations plutét qu'entre les séquences elles-mêmes – que nous souhaitons mesurer et utiliser pour la validation.

La comparaison des matrices d'autosimilarité requiert une normalisation de la longueur des séquences χ_a^b et χ_c^d é comparer, normalisation s'appuyant sur la fonction

optimale d'appariement des deux séquences. étant données les deux séquences normalisées de longueur l, $\tilde{\chi}_a^b$ et $\tilde{\chi}_c^d$, plusieurs métriques sont possibles. La plus simple consiste à prendre la norme l_1 normalisée, soit $D_{\text{SSM}}(\chi_a^b, \chi_c^d) = |\Phi(\tilde{\chi}_a^b) - \Phi(\tilde{\chi}_c^d)|/l^2$. Cette distance reste cependant très dépendante des valeurs absolues des éléments des matrices et ne refléte que peu la similarité visuelle. Afin de prendre en compte la structure spatiale des matrices d'autosimilarité, nous avons recours à une technique basée sur les histogrammes de gradients orientés Junejo *et al.* (2008)[1]. L'idée générale d'une telle approche est que l'apparence locale d'une matrice d'autosimilarité se caractérise bien par la distribution des gradients d'intensité locaux. Chaque matrice est ainsi transformée en un vecteur de caractéristiques locales, composé des histogrammes des gradients d'intensités pris localement en divers points. La distance entre deux matrices est alors définie comme la norme l_1 entre leurs vecteurs de caractéristiques et notée D'_{SSM}.

Les deux métriques D_{SSM} et D'_{SSM} apportent des informations complémentaires sur la structure des matrices d'autosimilarité. La premiére mesure directement la différence d'intensité entre les entrées de la matrice. En revanche, la seconde est invariante é l'ajout d'une constante à chaque entrée de la matrice. De plus, en ne se limitant pas à des informations ponctuelles, elle permet de prendre en compte une information plus complexe. En pratique, on utilisera donc en paralléle les deux métriques pour valider une répétition comme occurence d'un motif si $D_{\text{SSM}}(\chi_a^b, \chi_c^d) < \epsilon_2$ et $D'_{\text{SSM}}(\chi_a^b, \chi_c^d) < \epsilon_3$.

1.5 Résultats

Nous évaluons tout d'abord l'approche par DTW segmentale pour la découverte de mots dans un flux de parole avant de présenter des résultats préliminaires sur les distances D_{SSM} et D'_{SSM}.

1.5.1 Découverte de mots dans un flux

Nous avons artificiellement créé un flux de 10 h de signal par concaténation de dix enregistrements d'une heure chacun, dans l'ordre chronologique. Les six premiéres heures (2 h x 3 chaénes) ont été enregistrées sur une période de 15 jours, les quatre premiéres correspondant au même jour. Les quatre derniéres heures, provenant de

[1] Nous tenons à remercier émilie Dexter et Patrick Pérez qui ont aimablement mis leurs programmes à notre disposition.

4 chaénes différentes, correspondent à une période de 2 jours, éloignée de 18 mois de la premiére période. Le choix des données répond à deux considérations majeures. D'une part, on trouve de nombreux mots ou séquences de mots présentant à la fois des répétitions à court terme (au sein d'un reportage par exemple) et é long terme (reportage sur le même sujet mais sur une autre station le même jour ou le lendemain). D'autre part, nous disposons sur ces données d'alignements phonétiques permettant de faire correspondre les motifs découverts au niveau acoustique avec une transcription phonétique.

Dans toutes les expériences, le signal est représenté par des vecteurs de 12 MFCC, plus l'énergie, extraits à une fréquence de 100 trames par seconde.

La qualité des motifs découverts est évaluée au niveau phonétique. Rappelons que le résultat du processus de découverte de motifs est un catalogue de motifs, C_i, chacun caractérisé par ses occurences. La transcription phonétique permet d'associer à chaque occurence j de C_i sa transcription phonétique $C_p(i,j)$. Le motif C_i peut alors étre représenté au niveau phonétique par son centroéde, défini comme l'élément $C_p(i,j)$ le plus proche de toutes les occurences du motif. La précision d'un motif correspond alors à la proportion d'occurences suffisamment proche du centroéde. Le rappel est défini par rapport à l'ensemble des chaénes phonétiques suffisamment proches du centroéde de C_i dans la transcription phonétique du flux.

Pour découvrir des motifs correspondant à des mots ou séquences de mots, nous avons fixé la taille de l'amorce à 0,3 s et celle de la mémoire tampon à 120 s. Le seuil ϵ_1 a été réglé empiriquement de manière à obtenir un bon compromis entre rappel, précision et temps de calcul. Sur les 10h de signal, nous avons trouvé environ 3000 motifs, avec une précision de 85 % et un rappel de 25 %. Les motifs trouvés sont donc peu entâchés d'erreurs mais la DTW permet difficilement de grouper des occurences d'un motif qui présente une trop grande variabilité, expliquant ainsi le faible rappel. En particulier, la DTW est très dépendante du locuteur et les occurences d'un même motif par différents locuteurs ne sont pas détectées comme un unique motif mais plutét comme autant de motifs séparés. Augmenter le seuil ϵ_1 permettrait d'augmenter le rappel au prix d'une forte baisse de la précision. En effet, les motifs dans le catalogue sont représentés par la forme moyenne des occurences trouvées pour ce motif. Augmenter ϵ_1 engendre alors un nombre accru de fausses détections qui viennent déteriorer la représentation des motifs dans le catalogue.

De manière qualitative, les motifs trouvés correspondent principalement à des mots ou des courtes séquences de mots. Par ailleurs, plusieurs motifs sans contenu

Table 1.1: Précision/Rappel (en %) pour la détection de locutions clés dans un flux de 20 minutes

locution	D_{DTW}	$+D_{\text{SSM}}$	$+D'_{\text{SSM}}$
Jean Marie Le Pen	33 / 59	40 / 59	56 / 59
vingt-et-un avril	18 / 71	22 / 71	43 / 71
extrême droite	17 / 57	25 / 57	67 / 57
France	11 / 43	18 / 39	22 / 35

linguistique sont également trouvés. C'est notamment le cas des inspirations et des *jingles*.

Finalement, il convient de souligner que le temps de calcul pour le traitement des 10h de signal a été d'environ 13h. Méme si des optimisations permettrait de décroétre de manière significative le temps de calcul, ces chiffres mettent en évidence la difficulté du passage à l'echelle de notre algorithme dans le cas de la découverte de mots. En effet, la taille du catalogue de motifs croét rapidement pour ce type de données, ralentissant ainsi l'algorithme. Ainsi, nous avons mesurè que le temps de traitement en fonction du temps dans le flux est une fonction exponentielle (de la taille du catalogue).

1.5.2 Utilisation des matrices d'autosimilarité

Avant d'utiliser les métriques D_{SSM} et D'_{SSM} pour la découverte de motifs, nous les avons tout d'abord validées dans un cadre de recherche de motifs connus. Nous avons artificiellement construit un signal de 20 minutes par concaténation de six reportages sur le théme du 21 avril 2002, provenant de radios (et donc de locuteurs) différentes. Quatre locutions clés – Jean-Marie Le Pen, vingt-et-un avril, extrême droite, France – , caractérisées par une occurence de référence chacune, sont recherchées dans les 20 minutes de signal.

Les résultats, en terme de rappel et précision des occurences retrouvées, sont présentés dans le tableau 6.3. L'algorithme de DTW segmental présenté à la section 1.4.1 peut étre utilisé pour cette recherche (colonne 2), l'occurence de référence du motif à rechercher jouant le réle d'amorce. Les occurences trouvées pour chaque motif sont ensuite validées en utilisant la distance D_{SSM} (colonne 3), éventuellement complétée par D'_{SSM} (colonne 4). Ces résultats mettent clairement en évidence l'intérét

d'une mesure entre matrices d'autosimilarité pour la validation des motifs, permettant ainsi une amélioration substantielle de la précision pour un rappel constant (é l'exception du motif *France*, très court). Les occurences trouvées correspondent bien à différents locuteurs, tant masculin que féminin.

Des premiéres expériences sur l'utilisation des distances entre matrices d'autosimilarité pour la tâche de découverte de motif sur un autre extrait de 20 minutes confirment l'intérét de ces distances. En utilisant conjointement les deux distances, la précision augmente de 52 % à 66 % et le rappel de 42 % à 51 % par rapport à la seule DTW segmentale. Par ailleurs, l'analyse qualitative des résultats montre que des occurences du motif par différents locuteurs sont retrouvées pour certains motifs, comme *élevage* ou *poisson* .

1.6 Conclusion

Nous avons proposé une approche pour la découverte non supervisée de motifs sonores dans le signal de parole. La plupart des motifs retrouvés correspondent à des mots ou des séquences courtes de mots qui peuvent étre utilisés comme mots clés sonores pour caractériser ou indexer un signal. La méthode utilisant l'alignement temporel dynamique permet de detecter des mots clés avec une bonne précision mais présentent un rappel faible. La combinaison de l'alignement temporel dynamique avec la comparaison des matrices d'autosimilarité permet d'améliorer la découverte de motif au prix d'un effort calculatoire supplémentaire. Ce travail ouvre de nombreuses perspectives, tant pour améliorer la méthode que pour intégrer la découverte de motifs dans des applications d'indexation de documents oraux. En particulier, deux problèmes nous semblent cruciaux. D'une part, le passage à l'échelle reste problématique. Par ailleurs, afin d'utiliser efficacement les motifs découverts, il convient de les caractériser afin de ne conserver que ceux qui décrivent effectivement un contenu linguistique.

Chapter 2

Introduction

2.1 Motivation

Nowadays, multimedia data sets of massive size are available to a large audience of consumers worldwide. This results from multiple factors: affordable data generation devices, efficient compression techniques, high capacity storage devices and the pervasiveness of the Internet for sharing and spreading such information. Data (or *signals*) users can access, come in different forms: video, audio, image, text.

Extracting information from signals. When dealing with large collections of data, a popular issue concerns the extraction of documents pertaining to a user query. The query is usually expressed by a keyword in written form and the identification is done by matching it with a list of video or image tags, text documents, or transcribed speech data. Thus, regardless of the nature of the data set, the problem is merely reduced to a string matching issue. Not always, however, the document of interest can be concisely summarized by a specific word or a list of words: for example, suppose to search for a piece of music into a database of songs. Imagine the same scenario when searching for a visual pattern in a database of images, or a still shot into a collection of videos. On a related note, one might wonder how to cope with the possible absence of annotation, tags, speech transcriptions. Accurate annotation and tagging often require human intervention. When handling large data sets, this can be tedious, time-consuming and error-prone. Furthermore, transcribing speech into text requires the use of automatic speech recognition (ASR) systems: these systems are language dependent, imply a relevant computational effort to perform transcription,

and require collecting and annotating corpora on specific languages or topics in order to train acoustic and language models.

The need for effective ways to access and browse through these documents, has sparked intensive research in domains such as content-based multimedia indexing and retrieval. Two main issues lie at the core of these research efforts:

1. how to search a piece of signal in a collection of signals (databases or data streams), directly at the signal level (query by content).

2. how to extract *useful information* from the collection, at the signal level and in the absence of a specific user query.

The second problem is not trivial to characterize, if a precise definition of *useful information* is not provided as well. When browsing large data collections, one might want to skip redundant parts, focus onto specific parts, rapidly get a coarse understanding of the content, visualize data in a convenient way. Examples of tasks targeting such applications are several. Automatic text summarization aims at producing a non redundant extract of a text to be employed in several applicative contexts: in search engines, to present compressed descriptions of the search result, or in keyword directed subscription of news, which are summarized and pushed to the user (e.g.: *Web feeds*). In video processing, key frames are extracted with the aim of summarizing a video sequence by presenting a user, a selection of a few representative still shots; moreover, under the condition of heavy video data but limited storage capability, storing key frames merely, can be a comprehensible solution for data compression. Much like annotations and tags, key frames and text summaries provide useful information in the form of semantic cues on specific parts of the data set.

Problem definition. These issues and the underlying theoretical problems, are the main source of inspiration for the investigations detailed in this thesis. Here we tackle the problem of discovering occurrences of repeating audio segments in streams by unsupervised learning. As a matter of example, consider the picture in Fig 2.1, where the speech signal and respective spectrogram are reported for the French spoken expression: *la gauche est mal en point, le recul de la gauche est international.* What is demanded from the task is to automatically recognize, without any additional modality of information or a priori knowledge, the similarity of those two fragments marked by red rectangles, corresponding to occurrences of the word group *la gauche.*

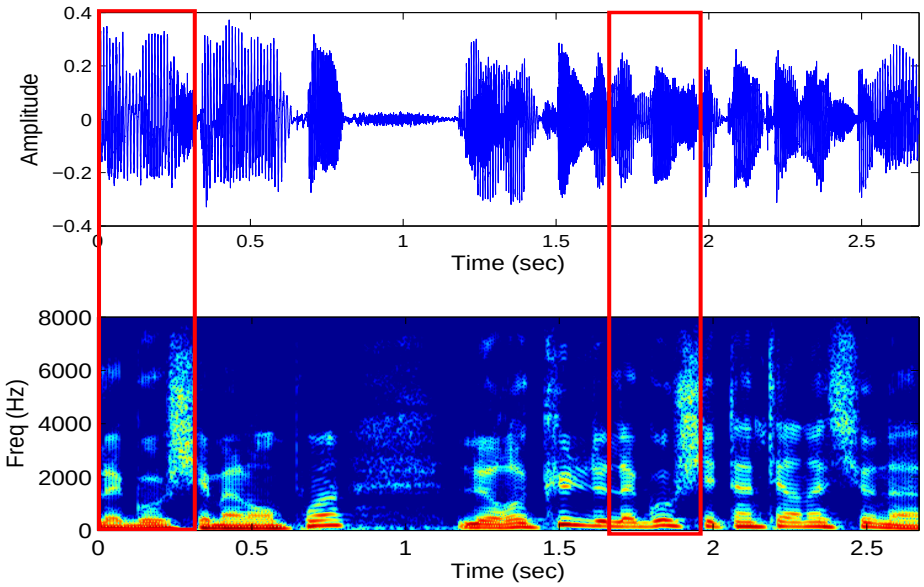

Figure 2.1: Speech signal and spectrogram for the expression: *la gauche est mal en point, le recul de la gauche est international*. The segments marked by red rectangles refer to occurrences of *la gauche*.

Automatic recognition of such similarities at the signal level is not a straightforward operation; and the task is even more difficult when minimal knowledge is available to be exploited for the recognition. *Knowledge* come in different forms: being aware of the presence of a repetition, knowing that the repetition concerns the word *la gauche*, disposing of explicit audio templates of that word, are all forms of knowledge that guide the recognition process, whenever they are available in advance. In our context, we assume that such knowledge is not available a priori for discovering similarities in audio.

In general, the underlying class to which similar occurrences belong to, is referred to as a *motif*. The term is borrowed from comparative genomics, where it designates a family of symbol sequences (each symbol representing a nucleotide or amino-acid). As biological motifs are allowed wild cards, similarly we do not focus only on identical audio segments, but rather on *similar* acoustical patterns for which a certain amount of variability is admitted. Speech signal is intrinsically variable, occurrences of a

word may differ significantly across different channels, speakers, speaking rates and environmental conditions. We do not impose any specific length constraint on the admissible audio motifs, if not for a lower bound, introduced to avoid motifs as short as a few samples of signal: we are therefore equally interested in repeating words, small multi-words phrases in speech data, or jingles, station call signs and signatures, songs, even entire movies or shows, in broadcast data. Motifs can repeat in a regular fashion or with no apparent regularity at all. Repetitiveness of audio patterns often reflects their semantic relevance within the data set: topic specific terms are likely to occur frequently. From this angle, audio motif discovery can be regarded as a potential counterpart task of textual keyword detection and key frame extraction.

Fundamental methodology. As far as the methodology is concerned, we approach audio motif discovery departing from the prevailing *train and test* paradigms for recognition. These methods model *a priori* the targeted patterns and learn parameters in a preliminary training stage, relying on an abundance of annotated training data. While this *supervised* approach has proven successful, the drawbacks that limit their attractiveness are also several: first, collecting and annotating large amounts of training data is often done single-handedly by human experts; next, a priori *modeling* implies to know beforehand the target of the discovery, a knowledge that often is not available and requires to be preliminarily acquired; third, training of models depends heavily on the quality of the training material, and might generate mismatch issues rising from the differences between the acoustics observed in the training and in the testing data.

While overcoming this limits might justify alone the need for a different approach, the motivation behind *unsupervision* is also more philosophical: it tries to determine to what extent a machine alone can learn from raw data, in the absence of any prior or side knowledge (in the audio field, *acoustic* and *linguistic* knowledge). It can be regarded as a sort of lower bounding methodology to machine learning and understanding of audio.

Audio motif discovery is therefore based on the following principles:

- no training data is used
- no additional source of information (text, video, image, external acoustic data) aids the discovery task

- language models are not used, while the acoustic ones are learnt and refined as motifs are discovered

In the proposed framework, this general philosophy translates practically into an incremental learning framework where everything can be learnt on the fly, as the incoming audio stream is received and processed.

2.2 Related work

The adoption of unsupervised strategies is an emerging trend in audio information retrieval tasks; that explains why most related work in audio motif discovery dates back to this last decade.

It is possible to roughly characterize existing work based on the nature of the targeted motifs: in *word* discovery, the goal is to find occurrences of repeating words in speech data (Park (2006); ten Bosch & Cranen (2007)); the complementary task, on the other hand, aims at discovering signalling patterns in multimedia streams from audio, like repetitions of jingles, advertisements, songs, that will be referred to, throughout the manuscript, as *near-duplicate* patterns.

While it is easy to recognize that the underlying task is the same, various approaches have been proposed, mainly to deal with the very different degree of variability and length of the sought patterns: words are short and variable, near-duplicate patterns are longer and (almost) identical. In a similar way, researchers dedicated to these two tasks have often come from different domains: the speech community in the word discovery case, and the multimedia community in the other one.

We provide in the following an overview of existing work in both tasks.

2.2.1 Word discovery

Park: Unsupervised word discovery in speech. A pioneering study in word discovery is represented by the doctoral work of Park (Park (2006)) and related publications (Park & Glass (2005, 2006, 2008)). Park employs a batch processing of a speech file that involves four different steps to perform the complete task: a) segmentation of speech data into smaller fragments separated by silent intervals, b) identification of similarities between fragments by a pairwise comparison based on a segmental version of Dynamic Time Warping (SDTW), 3) production of an adjacency graph, with pairs of matching segments as nodes connected by weighted edges, the

weight being the respective DTW score and d) clustering of nodes to group motif occurrences.

The architecture is depicted in figure 2.2: the segmented utterances (indicated by an integer index) undergo a pairwise comparison, generating a set of triples. Each triple comprises the endpoints location of the two matching audio segments within each utterance, and the distortion score of the respective alignment path. The node extraction procedure serves to label with a unique discrete time index, audio segments among different triples that overlap in time (and are indeed recognized as referring to the same pattern). For the sake of clarity, consider the step b) and c) in figure 2.2: matching subsegments in fragment 1 and 3 resulting from the three comparisons are strongly overlapping and belong indeed to the same occurrence, hence labeled by the same time index; the two subsegments of fragment 2 do not significantly overlap, and refer to different patterns, hence labeled differently. Once the time indexes and edge weights are available, each edge being the distortion score $w_{i,j}$ in figure 2.2, indexes and respective patterns are further grouped into clusters. The clustering procedure, borrowed from Newman (2004), permits to include the edge weight information and groups occurrences in a greedy fashion, privileging more densely connected nodes as cluster members.

Very recently, the follow-up paper (Zhang & Glass (2010)) explicitly addresses speaker dependency issues deriving from the adoption of acoustic features like MFCC. To handle inter-speaker variations in speech, the use of Gaussian posteriorgrams is proposed, obtained from models trained on a large corpus of multi-talker data. Our approach to the problem is to avoid, as much as possible, resorting to external data and training tasks before the actual discovery.

ten Bosch: A computational model for word discovery. Conceptually similar to Park's work is the computational model proposed in ten Bosch & Cranen (2007), in that the discovery is achieved by a combination of speech segmentation, clustering, and temporal sequence learning; precisely, in the experiments described, the acoustic sequences of a digit-string database are segmented into *phone-like* fragments (that is spectral-homogeneous sequences of acoustic vectors). These fragments are grouped into clusters according to a DTW distance-based k-means procedure. Thanks to this quantization of phone-sized sequences, the original digit-strings are converted into sequences of integers. The likelihood of these sequences sharing a common word is determined by a pairwise comparison based on local alignments by modified DTW

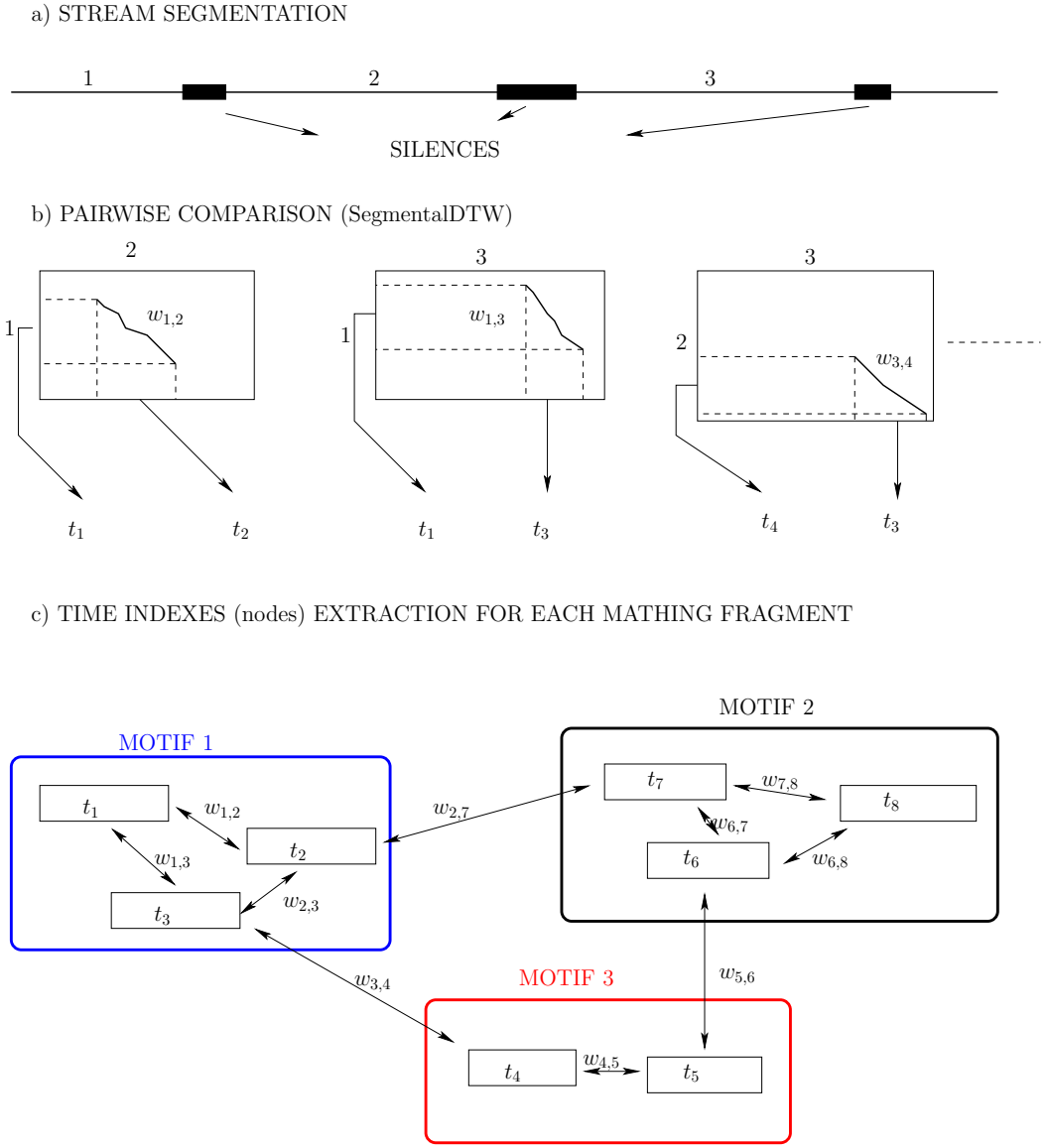

Figure 2.2: Main steps comprising Park word discovery system.

(much like stage two of Park's framework, but over sequences of integers in this case). In a final stage the detection of matches is assisted by the use of tags that indicate the presence of a specific word (in fact, a number) within the utterance, but not its exact location nor its acoustic realization. This additional information roughly

reflects the role played by the visual modality in language learning. In fact, visual and spoken co-occurrences of patterns assist the young infant in acquiring the units of language. For example, the tag *yes* associated to the utterance *look at this nice ball*, flags the presence of the word *ball*, though the machine is unaware of the explicit relation between the tag and the word. In this context, the tag mimics the visual appearance of a ball when the utterance *look at this nice ball* is effectively spoken, while the language learner is unaware of the association between the object and the corresponding spoken word. Even though acoustic repetitions are unknown as in our assumption, here discovery is explicitly assisted by an additional modality of information, and is not performed over a continuous stream of data, but rather on isolated spoken sentences.

Stouten: Automatically learning the units of speech. Somewhat related with our task is the work in Stouten *et al.* (2007). Rather than words, (word-sized) phone patterns are discovered that are present in speech utterances. Discovery is performed by applying Non-negative Matrix Factorization (NMF) to a high dimensional representation of speech utterances derived from a set of phone lattices, and by identifying the basis vectors of the decomposition as the structural units of the spoken sentences. More precisely, a database of t digit strings is represented by a matrix $\mathbf{V}$ of size $n \times t$ each column i being a sequence of weighted co-occurrence counts of phone pairs in the i-th sentence (thus n is the square of the number of phones in the database). An approximate factorization of $\mathbf{V}$ is computed by NMF under the (fulfilled) constraint that all entries are non-negative, in the form:

$$\mathbf{V} \approx \mathbf{WH} \tag{2.1}$$

where $\mathbf{W}$ has size $n \times r$ and $\mathbf{H}$ has size $r \times t$. Usually the value of r is chosen so that $r(n + t) < nt$, so that the reconstructed matrix has a reduced dimensionality. $\mathbf{H}$ and $\mathbf{W}$ are computed by iteratively updating rules that seek to maximize an appropriate objective function. What equation (2.1) tells is that each column $\mathbf{V}_{:j}$ (each data vector) can be expressed as a linear combination of the columns $\mathbf{W}_{:k}$ (the fundamental speech units) weighted by the coefficient of $\mathbf{H}_{:j}$. The authors perform an experiment where r is set equal to 11 (the number of different words comprising the sentences: the integers from zero to nine plus a silence) and show a high quality of approximation of the matrix $\mathbf{V}$, which dramatically plummets for $r < 11$ and slightly increases for $r > 11$. Moreover, they show how the columns of $\mathbf{W}$ are sparse vectors

whose dominant values correspond to those phone pairs that actually occur in the digits. This provides empirical proof that the input data can be roughly described as a combination of a few words (or sequences of phone pairs occurring in those words) that are the ones effectively repeating.

This work is related to ours in the sense that the actual discovery is done by unsupervised learning (through NMF) and the patterns found are effectively repeating. However, the representation of speech utterances is generated using an acoustic model and a bigram model that are preliminarily estimated on training data (the Wall Street Journal database); the presence of a training stage already differentiates this work from our approach (even if it involves the feature extraction process rather than the learning paradigm). In addition, the patterns found are rather phone pairs occurring in words rather than words. Moreover, the knowledge of the phone identities and their numbers are required, and the discovery is performed on isolated sentences rather than on a continuous data.

Jansen: towards efficient discovery of long words. Very recently, a novel technique for discovering repetitions in speech has been proposed in (Jansen *et al.* (2010)). Here the length constraint on the targeted patterns is more stringent, as discovery is limited to speech segment of at least one second of duration, under the assumption that longer terms are somehow more contentful and relevant. The search for repetitions is performed by a) representing the search space as the self-distance matrix of the speech file and b) adopting basic image processing procedures to identify diagonal lines (the patterns induced by two matching segments) within an image (the distance matrix). Since the complexity is quadratic with the file length, to achieve a significant speed-up in distance matrix computation, the use of language-specific posteriorgrams is exploited to map acoustic features (MFCC) into sparser vectors. Pattern discovery is further refined by applying Park's SDTW to the hypothesized matches, to account for deviations from the straight line, likely due to natural prosodic variations in speech.

Anguera: subsequence template matching in speech In Anguera *et al.* (2010) a modified DTW algorithm (U-DTW) has been described to enable the detection of matching speech feature subsequences in a pairwise comparison. The approach does not differ in principle from the similarity detection subtasks in Jansen *et al.* (2010) and Park (2006): a distance matrix is built from the pair of feature sequences, and

the search for the matching area is conducted by identifying a low-scored alignment path within the matrix. The main difference with the aforementioned procedures resides in the practical strategy implemented for performing the detection: a division of the matrix in uniform regions (either horizontal or diagonal) is carried out, whose size is related to the minimum length of a possible word occurrence. According to such division, alignment path are computed from starting points and confined within the respective matrix subregion. A path extension heuristic is then applied to allow for paths to grow in an unbounded fashion and reconstruct a matching path in its entirety. This specific technique will be shown to exhibit notable similarities with our own modified DTW procedure, in the more technical part of the manuscript.

2.2.2 Near duplicate discovery

Concerning the discovery of signalling patterns in composite broadcast audio, two main works are mentioned as particularly representative, respectively Herley (2006) and Lu & Hanjalic (2009).

Herley: identifying repeating objects in multimedia streams In Herley (2006) repeating objects are discovered in multimedia streams by relying on the sole analysis of the audio signal. The system proposes the use of sequential processing and low-dimension fingerprints of the signal to accomplish the task. Sequential processing is adopted as a requirement for handling (possibly infinite) streams of data. This implies that the search for repetition is performed by relying only on the received and available portion of the data. In practice, the discovery is organized by searching a fixed length fragment of audio either into a collection of repetitions already found (an incrementally built and updated catalog of repetitions), or, if not there, in the past data stream already processed. As far as pattern recognition is concerned, the time correlation of audio fingerprints is computed as indicative of the (dis)similarity of compared patterns. The fingerprint is obtained by only retaining the sixth of the 25 Bark bands the audio signal can be splitted into. These are frequency selective channels that can be sampled at a much lower frequency than the overall audio signal. In the described experiments, Herley reports good results in discovering patterns of the duration of several minutes by using a distorted version of the signal sampled at a rate of 11 samples per seconds (thus obtaining a 4000-fold reduction in dimensionality with respect to the original signal, sampled at 44KHz). We will come back later on to this system, that will be shown to have important implications on our work.

Lu: Audio content discovery In Lu & Hanjalic (2009) the goal is not that of discovering repetitions in multimedia streams, bur rather of extracting a set of semantically meaningful audio elements to describe the content of the composite audio stream. Composite audio implies that the stream is populated by a multiplicity of audio modalities: speech, music, laughter, applauses, various sounds either mixed or following each other in a sequence. We can roughly attempt to summarize this work by listing its three main steps:

1. stream segmentation into pieces of audio and their clustering into respective sound classes (speech, music, etc...), called *audio elements*. This step is practically performed by appling spectral clustering to a representation of the data made of temporal features (short time energy, zero crossing rate) and spectral features (sub-band energy ratios, brightness, bandwidth, 8-order MFCCs). This step comprises already a discovery system, as it groups in an unsupervised way audio segments according to cluster membership (the cluster being a semantic class of sound, rather than a motif).

2. among these *audio elements*, key segments are spotted in terms of semantic relevance, according to a score that accounts for element duration, frequency of occurrence, average length and average length variation. This is because, depending on the sound, semantic importance might be defined by different values of these factors: for examples, unusual sounds in surveillance videos do not occur frequently and are short in duration, however constitute a key target sound in that context. Applauses in a tennis game, or laughters in a situation comedies are expected to be also frequent, and have roughly similar length in each of their occurrences.

3. Scene categorization by information-theoretic co-clustering, which is achieved by exploring co-occurrences of key segments of different audio elements that occur simultaneously in a common auditory scene. For example, *gun shots* and *explosions* belong to different sound classes, but their co-occurence in a *war scene* may help discovering the higher-level semantic class they share.

As evidenced by this summary, Lu's work is not specifically intended to cluster repetitions of similar patterns, but rather to cluster patterns in classes (according to the nature of the acoustic realization, or to a high level semantic concept). On the other

hand, it shares with our framework, the fundamental methodology that seeks to extract information in a completely unsupervised fashion, without relying on training data or a priori knowledge.

2.3 Potential applications of audio motif discovery

Audio motif discovery, in particular the word discovery case, is an emerging topic, based on novel paradigms, within the speech and audio community. Being in the early stage of its research history, most applicative tasks targeted by the theoretical issue are rather hypothesized and suggested, than explicitly shown in practical scenarios. We attempt to briefly present such applicative frameworks before proposing additional ones in chapter 9.

In the introductory part of the chapter, the possible analogy has been drawn between audio motif discovery and keyword extraction in text documents or key frames in video sequences. The underlying idea is that repetitive sound patterns tend to convey semantically relevant information on the data content. While we do not claim that this is actually true for all the motifs in a data set, these patterns can be used at least as an input to some strategy aimed at selecting them according to a specific score of semantic relevance. Indeed this is proposed in the described work of Lu & Hanjalic (2009), where a score is appropriately defined for spotting key occurrences within clusters of audio elements. In regard to word level key occurrences, a recent work explicitly addressing this issue is Zhu *et al.* (2009). Here multiple spoken-document summarization is performed by first discovering acoustic re-occurrences by Park segmental DTW, then by progressively extracting keywords from this set of utterances, according to a maximum marginal relevance criterion, that seeks to maximize the balance between salience and redundancy.

In Park & Glass (2005); ten Bosch & Cranen (2007) supervision and independence from training data are seen as the possible novel paradigm for liberating speech processing by the mismatches occurring between the acoustics in training data and in the test data, within the traditional supervised learning paradigm. The out of vocabulary problem (OOV) is one example of such mismatch, in this case between the employed lexicon of word units and the actual words in the data. Audio motif discovery does not suffer such problem, as a lexicon of units is inferred directly from the test data,

and, for example, might somehow be integrated as a recovery mechanism of OOV words in a modern ASR systems.

On a different note, in the last few years, work carried out within the project ACORNS[1] has studied unsupervised methods for learning co-occurrences of spoken patterns, in the attempt to mimic the mechanisms that allow young infants to acquire the units of a language, in the absence of a priory knowledge. This investigations take inspiration by advances in developmental psychology that suggest the crucial role of recurrence of patterns in the early stages of language acquisition. In (Saffran (2002); Saffran *et al.* (1996)) an experiment is described where 8 months old infants were exposed to a continuous stream of speech, generated by concatenating repetitions of four three-syllables word entities and random sequences of syllables. After only two minutes, infants showed a remarkable capability of discerning the repeating word entities from the random ones, in the absence of prosodic or acoustic cues for boundary detection.

Concerning the discovery of long signalling patterns in multimedia streams, Herley enumerates several attractive applications of the technique in Herley (2006):

- broadcast monitoring: by detecting repeating objects and their locations, it is possible to verify that a pattern (for example, an advertisement) has been played when expected.
- commercial skipping and stream customization: the construction of a library of repetitions might enable to remove all the occurrences of the unwanted items (like commercials within a movie) to be replaced with alternative material.
- statistics gathering on broadcast data: the collections of such repeating patterns allows to track the distributions of play frequencies of patterns in real streams and other related statistics.

Besides the ones listed in this paragraph, in Chapter 9 new ideas will be proposed on the possible employment of audio motif discovery in applicative scenarios.

[1] http://www.acorns-project.org/

2.4 Motif discovery in other domains

While the theoretical problem and the associated methodology represents essentially a novelty within the audio and speech community, in other domains, unsupervised discovery of patterns is indeed a well established research topic since many years.

Computational biology In comparative genomics, among others, the discovery of recurrent subsequences, hidden inside large sequences (of DNA or proteins), serves to identify regions that are believed to play key roles in biological functionalities. Since the extremely high number of work in this field, we remind a couple of tutorial papers in (Brazma *et al.* (1998); Sandve & Drabløs (2006)). Curiously, while our work insists on the passage to unsupervision in discovery, the trend seems inverted in biological sequence analysis, as knowledge on data is becoming more and more available. By taking advantage of the immense amount of data produced by large-scale DNA sequencing efforts (such as the Human Genome Project), the current approach mainly relies on probabilistic modelling of sequences and statistical estimation of model parameters (see Durbin *et al.* (1998)).

Data mining In the last decade, the topic has received increasing attention also within the data mining community. In Lin *et al.* (2002) the discovery is performed on a symbolic representation of the data, obtained by dimensionality reduction and discretization. In Minnen *et al.* (2007) discretization is not used. Instead each motif is initialized *online* by estimating parameters of a hidden Markov model (HMM) from subsequences located near the density modes (local maxima) of the distribution of the data in the feature space. The model is then used for detection of further occurrences, similarly to what is done in a HMM-based ASR system (see Rabiner & Juang (1993)).

Pattern extraction in music information retrieval Another related line of research comprises what, in the music analysis community, is often called audio thumbnailing or snippet extraction (Burges *et al.* (2005); Chai & Vercoe (2003); Dannenberg & Hu (2002); Goto (2003); Logan & Chu. (2000); Peeters *et al.* (2002)). Often the common framework consists in converting a piece of music into a feature sequence used to build a self distance matrix. This matrix is then processed to infer music structure, summarize music files, detect duplicate music files, etc. The use of the distance matrix will also be exploited in the pattern matching techniques detailed later in the thesis.

Robotics An example of application in robotics is provided by the work in Lattner & Herzog (2004). A learning approach is described that learns temporal patterns in a sequence of predicates within a top down induction framework. It targets the goals of making agents more flexible to adapt their behaviour to the surrounding environment. This is necessary as agents take decisions and perform actions according to an interpretation of scenes and situations that match with their current belief of the world. Much like the OOV problem in speech recognition, contexts outside the defined set of actions and decisions impair the capability of the agent to handle properly the encountered situation. In this case, pattern acquisition by discovery, is seen as a countermeasure to allow the agent to autonomously deal with unexpected situations.

Association rules mining In Hoppner (2001) pattern discovery is used rather as a subroutine to infer temporal rules, in the spirit of association rules mining. Basically, a series of labeled intervals is processed to find recurrent temporal patterns, defined as sets of states and respective interval relationships, such as *A before B, A overlaps C, C overlaps B*. As an applicative case, a relationship is shown between air-pressure curve and wind strength, based on the inferred local weather forecasting rules.

Multimedia Finally, Xie (2005) proposed pattern discovery to induce high-level semantic concepts as break or play segments in sports matches in multimedia streams using a graphical modeling approach. Here, the discovery task amounts to estimating the optimal parameters of a hierarchical HMM, where the recurring patterns are modeled as HMMs linked to each other via transitions in a higher-level Markov chain. The experiments were conducted on broadcast data using features such as dominant color intensity and motion intensity for video, and volume, zero crossing rate, and spectral roll-off for audio.

2.5 Claims and Contributions

The primary achievement of this doctoral work has consisted in the construction of a computational architecture that discovers and collect occurrences of repeating acoustical patterns in an unsupervised fashion. The result is a framework that processes data in a sequential manner to deal with streams. The system is, at least partially, successful in dealing with the high variability of speech for finding repeating words,

and is able to deal with large streams of broadcasted data for finding repetitions of songs or advertisements. One major feature that we claim as distinctive of our system, resides indeed in its broad applicability to these related, yet significantly different, discovery tasks.

Beside the complete system, single propositions that we claim as original contributions of our work are summarized by the following items:

- we propose a formulation of the general discovery problem, and suggest a decomposition in subtasks that help understanding and solving the global task. Previously proposed systems are also shown to fit this modular structure, and their elementary components are identified.

- A general framework, that we call *seeded* discovery, is introduced that permits to discover and retrieve motif occurrences, regardless of their length and specific location in the data stream. This basically consists in detecting first a fixed length fragment of a motif from which the final occurrences are grown by their entire length through a match extension procedure.

- A template matching technique is introduced in the explicit attempt of improving robustness to variability with respect to DTW-based sequence comparison. This pattern matching technique is based on the comparison of self similarity matrices of speech sequences. A practical method is described for comparing and quantifying the degree of similarity of these matrices, adapting an image processing technique, widely employed in object classification tasks in computer vision. The benefit of the technique is evaluated in word spotting and word discovery experiments.

- An evaluation framework at the phonetic level is proposed, in order to assess rigorously the performance of the algorithm. The main contribution is the definition of novel measures of precision and recall for assessing the results of word discovery experiments.

- Variations of the popular DTW procedure are described that enable subsequence matching of speech sequences, by relaxing the boundary constraint of the classical algorithm.

- The concept of *audio icon* is introduced, and defined as an instance of a recurrent pattern in audio data. In general, recurrence of states is a fundamental

property of many natural processes and dynamical systems, and describes structural properties of the underlying process. Similarly in audio data, while repetitiveness does not necessarily implies relevance in terms of *semantic* content, it conveys information on how data is *structured* and organized.

2.6 Outline of the manuscript

The manuscript is organized and structured in the following manner: in Chapter 2 a formal definition of the problem is provided, and a decomposition in subtasks is proposed. Next, a number of basic concepts will be introduced that form the background for the presentation of a first motif discovery system. Such system will be illustrated in chapter 3, together with novel variations of the well known dynamic time warping algorithm, to find matching subsequences between pairs of spoken utterances and deal with the absence of an a priori segmentation of the stream into word units. Building upon this system, a more definitive architecture, seeded discovery indeed, is presented in Chapter 4, that will form the basis for the experimental evaluation detailed in the next two chapters. The word discovery case will be first considered, in Chapter 5, and additional template matching technique operating on the self similarity matrices of speech sequences will be also introduced, to improve robustness to speech variability. Next, seeded discovery will be used to address the task of retrieving longer and less variable repetitions, like songs, in radio broadcast data, by incorporating a number of additional features, mainly directed at properly handling large-scale issues. On a different note, in Chapter 7, we will insist on the applicability of motif discovery in audio mining tasks, defining the concept of audio icon, and highlighting its potential role in novel, more complex mining applications. The manuscript ends by presenting a summary of the work, as well as a number of hints for future development of the work.

Chapter 3

Problem statement and basic concepts

This chapter first introduces the task of motif discovery in a formal way. Next a decomposition into autonomous and separated subtasks is proposed as a way of approaching the global task in a systematic manner. Four subtasks are identified and carefully illustrated as a) segmentation b) feature extraction c) similarity detection and d) similarity score. Discovery systems previously proposed, are shown to fit this modular structure and the different elementary components are identified according to this paradigm.

Following the local progression implied by this decomposition, we deal with each subtask independently, in our pursuit of a computational system for motif discovery. Motivated by the goal of designing a streaming algorithm, the ARGOS framework (Herley (2006)) is reviewed in detail, with regard to the segmentation subtask. As a feature extraction technique, we resort to the classic representation of audio by mel frequency cepstral coefficients (MFCCs), widely employed for the description of sounds of various nature. The last two subtasks, similarity detection and score, are introduced by detailing the well known dynamic time warping (DTW) procedure for time aligning speech sequences and deeming their similarity. The presentation of such a popular technique is instrumental in introducing three variations of DTW, aimed at overcoming some drawbacks deriving from the boundary constraint and normalization strategies of the classical algorithm. Since this is where the most novel part of the initial system resides, the next chapter is entirely devoted to the presentation of such techniques and their integration into a discovery system.

3.1 Problem formulation

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

$$H(\chi_a^b, \chi_c^d) < \epsilon \tag{3.1}$$

$$|b - a| > L_{min} \tag{3.2}$$

$$a < b < c < d \tag{3.3}$$

Condition (3.1) represents the similarity condition; it formally states that two sequences are similar if their distance, measured by the metric H, is sufficiently small, i.e., below an appropriate threshold ϵ. Condition (3.2) imposes a constraint on the motif minimum length. It is reasonable, if not necessary, to restrict the discovery to sufficiently long patterns, discarding matches as short as a few samples of signal. The third condition prevents considering overlapping segments as occurrences of a same motif (obviously, the two segments obtained by slightly shifting one of them are very similar, but refer indeed to the same portion of data).

According to this formulation, the problem is reduced to that of separating the repetitive part of the data from the non-repetitive one, by grouping similar segments in a pairwise fashion. A subsequent post processing stage is then needed to merge clusters belonging to the same motif. This suggests to effectively view motif discovery as a clustering technique that operates only on that part of the data that effectively repeats. The scope of this problem statement is rather general: it applies for any kind of temporally ordered set of events, be it continuous or discrete, scalar or vectorial, regardless of the specific strategy adopted to solve the task.

Building from these considerations, we approach the problem by identifying the basic complementary subtasks, and dealing separately with each, in a modular fashion. The fundamental concept is that the appropriate decomposition help better understanding the problem itself, easing the proposition of a solution or the integration of different features into the same general framework.

3.2 Decomposition in subtasks

Audio motif discovery can be conceptually decomposed into four main elementary components:

Segmentation. The term segmentation here refers to the general strategy used to scan and process the data, according to some desired requirements. For instance, the choice between batch or sequential processing is a key element of this subtask. This choice has important implications on the performance and the computational resources demanded by the system. The term segmentation is employed as the processing strategy determines how the data is broken into segments to allow the search for repetitions.

For example, in (Park (2006)), this subtask can be identified in the macro-segmentation of speech fragments separated by silent intervals. These fragments are the acoustic units fed to the pattern matching procedure in a pairwise fashion, each against all the other ones. Here the segmentation subtask includes both the silence-based segmentation and the strategic choice to compare them all, implying a quadratic complexity with respect to the number of produced segments.

In Herley (2006) the segmentation is performed by breaking the data stream into a pair of adjacent audio segments, a current query and its recent past, in an iterative scheme that enables sequential data processing and avoids the quadratic complexity of *naive* approaches.

Feature extraction. It refers to the parametrization used to represent the data in a specific domain, suited to perform the discovery. It strictly depends on the nature of the data and on the targeted motifs. Hence, a solution is specific to a particular domain and task. In data mining, though, the common approach is to design algorithms and representations valid for a variety of data types, as long as they can be modeled as a time series of real-valued variables. According to this approach, the original data set is converted by an appropriate procedure in a discrete, symbolic representation where the task is performed. Usually this alternative representation is achieved by direct quantization, or by dimensionality reduction techniques. For instance, vector quantization of speech spectral vectors is adopted in Rasanen *et al.* (2009a,b); ten Bosch & Cranen (2007).

Various dimensionality reduction techniques are used in data mining work on time series: for example, in (Lin *et al.* (2002, 2003); Minnen *et al.* (2006)) a data representation called *Piecewise Aggregation Approximation* is used (Keogh *et al.*; Yi & Faloutsos (2000)) that consists in dividing the data set in equal sized frames, where the mean value is computed and then quantized; in (Tanaka & Uehara (2003)) reduction is achieved by principal component analysis.

In closely related work, like that of (Park (2006)), word discovery task is performed directly on the spectral vector sequences. We will follow the same choice throughout all our experiments. This choice will be later justified in Section 3.4.

Similarity detection. This task can be identified with the actual techniques employed to detect similarities in data. It explicitly refers to the pattern matching methods for selecting the segments most likely to be occurrences of a same motif (*motif candidates*). For instance, in Herley (2006) it corresponds to calculating the time correlation between fingerprints of audio segments; in Park (2006) it amounts to time aligning speech sequences by SDTW.

Similarity score. It is indeed equivalent to the computation of the metric H in (3.1). It is the score used to deem whether two segments (previously identified as motif candidates) are similar or not.

In Park (2006) this score is given by the cumulated distance of the minimum-weighted alignment path (it is, indeed, the DTW dissimilarity measure). The alignment paths found by the same Park's SDTW are scored differently in Zhu *et al.* (2009), by keeping into account also the length of the path and its degree of warping. In Herley (2006) this score is represented by the minimum value of the time correlation between audio segments.

Note that these last two tasks (i.e. similarity detection and score) can be easily integrated into the same one, depending on the specific solution proposed as the value of H may directly relate to the pattern matching technique used for similarity detection. However, the two tasks remain conceptually different. Indeed, one can imagine to employ an approximate pattern recognition technique to rapidly identify likely repetitions and then validate the hypothesized occurrences by performing a more accurate comparison. We also will see an example of this when the comparison of self similarity matrices will be introduced in section 6.4, as well as the use of a low and full resolution comparison in chapter 7.

We start the description of our (preliminary) motif discovery system by considering in detail each of these subtasks, motivating our choices with respect to our assumptions and objectives.

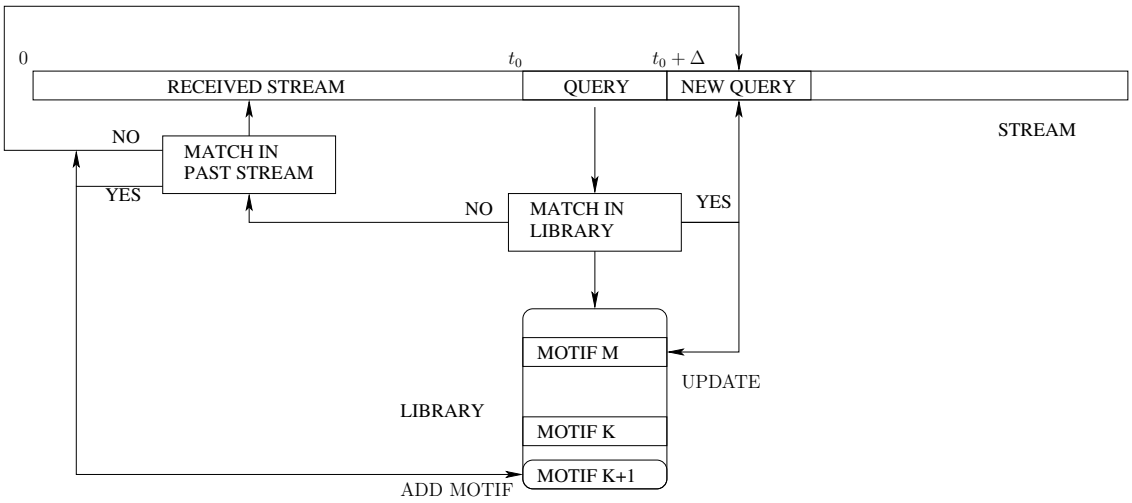

Figure 3.1: Algorithmic view of the ARGOS segmentation framework.

3.3 The ARGOS segmentation framework

A naive approach for finding repetitions, consists in considering all possible segments of admissible length (queries) and search each of them exhaustively in the data set. This strategy is unfeasible even for a small data set, as it implies a combinatorial explosion of search operations. Moreover, it requires all data to be stored and accessible, as each possible query is searched for in the entire file.

To overcome some of these drawbacks, we have resorted to the ARGOS segmentation framework proposed in Herley (2006). It is an approach that allows a sequential processing of the data, thus suited to process streams, and that exploits statistical properties of real audio streams to *smartly* reduce the search space. The discovery is performed by iteratively searching a sliding query of fixed length either into an incrementally updated collection of motifs (*library*) or over the past received portion of stream.

The algorithmic process can be illustrated in more details with the help of the diagram in Fig 3.1, where a single iteration is depicted. To improve clarity, we decompose the system into *objects*, *actions* operating on these objects, and *hypothesis*:

Objects

- **query**: it is the portion of the data stream indicated by the endpoints $[t_0, t_{0+\Delta}]$ in Fig 3.1. In streaming modality where data is progressively received, it is (part

of) the current input data received. The query is the segment to be searched at each iteration, assuming to be a repeating pattern itself or to include a repeating pattern as a portion. The search space where the presence of a repetition is determined is represented by a *library of motifs* and the *past stream* already received.

- **library of motifs**: it is the catalog where motif occurrences are collected whenever they are discovered. It represents the long term memory of the repetitive part of the past stream. It serves a double purpose: it is a way of *storing* the results of the discovery for each motif. But most importantly, it serves for the recognition of subsequent occurrences of the same motif, whether a specific modelling strategy is adopted or the complest list of current occurrences is used for the recognition.

- **past stream**: it is the portion of data stream already received and processed, marked by the endpoints $[0, t_0]$.

Actions

- **library search**: each query extracted from the stream is first searched into the library, by comparison with each motif therein. Library scan is stopped as soon as a match is found; in this case the library is updated by signalling the presence of a new occurrence for that motif (the motif M in Figure 3.1).

- **past stream search**: if a repetition of the current query is not found in the library, the search for a possible match is then performed in the entire past data already received and processed. Since it represents the search space where we attempt to find a repetition of the query, it will be often referred to as **search buffer**. If a repetition is detected a new motif is discovered, and an apposite entry is created in the library (the motif $K + 1$ in Figure 3.1).

- **query shifting and iteration**: whether or not a repetition is detected (in the library or in the past stream), a new query is extracted from the stream and the search operation is iterated. As shown in Figure 3.1, the new query is adjacent to the preceding ones, defined by the endpoints $[t_{0+\Delta}, t_0 + 2\Delta]$.

Hypothesis

- **local repetitiveness**: in real streams repetitions of objects occur within a limited time span, that can be reasonably predicted according to the specific task and targeted motif. For instance, repetitions of topic specific terms in a news show are expected to occur frequently within a few seconds or minutes; a song might be played a second time after several hours, in the broadcast schedule of a radio channel. Based on this assumption a speed up can be achieved by limiting the search in the stream to the most *recent* part of the received data. The local repetitiveness condition needs only to be satisfied once for a motif to be discovered. The discovery occurs indeed whenever two instances of a motif are detected for the first time; the subsequent occurrences are then identified by comparison with the respective motif in library search.

The restriction of the search space implied by the local repetitiveness assumption, rises two main considerations:

- the assumption that motifs are local (at least, once) can impair the recall performance of the system (that is its capability of collecting as many occurrences as possible for each motif). A motif can be discovered whenever two of its occurrences are reasonably close in time (no more distant than the recent past length); hence, the past occurrences that are not locally repetitive are definitely lost.

- setting an appropriate length for the recent past is reasonable, given the task; however it might be seen as a violation of the unsupervised paradigm, as it effectively amounts to estimating a priori the motif average frequency of occurrence for the given task.

In Fig 3.2 a comparison between the naive approach and ARGOS segmentation is illustrated: in the first scenario, the search space is represented by a self-distance matrix of the acoustic file. This representation emphasizes the quadratic dependency by the file length implied by this procedure; in ARGOS, a query (the current input) is searched in the library of motifs and in its recent past: the complexity of the search linearly depends on the library size and the recent past length. Besides, the sequential processing allows for a streaming algorithm, contrary to the naive method.

A crucial aspect concerns the choice of the query length and the relative position of subsequent queries: as mentioned, a naive but optimal method consists in considering

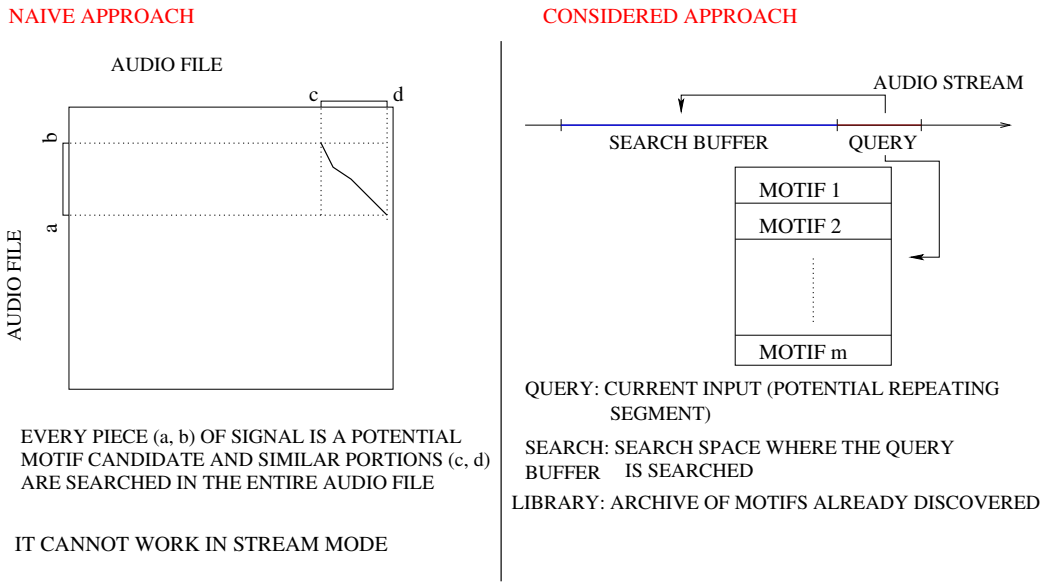

Figure 3.2: A visual comparison of the *naive* and ARGOS approach.

as a queries all possible audio segments of admissible length. This guarantees that a repeating pattern is extracted and searched, whatever its length and its location. As pointed out, this approach is not feasible because of its combinatorial complexity and the unavailability of the future data in a streaming mode framework, that prevents considering queries stretching in the future. Since the choice of the query length is also related to the particular pattern matching technique employed, we postpone to elaborate on this aspect to the similarity detection subtask. Before that, the parametrization of the audio signal will be discussed in the next section.

3.4 Feature extraction

The type of parametrization of the audio signal has been decided according to our ultimate goal of designing a system capable of dealing with different motif discovery tasks. Features should be sufficiently powerful and accurate to discriminate audio segments at the word level, but also suited to describe composite audio (speech, music, various sound effects that are mixed or follow each other in a sequence).

These requirements led us to resort to the classic mel frequency cepstral coefficient (MFCC) representation of the speech signal. The mel-frequency cepstrum is a

representation of the short-term power spectrum of a sound: it is based on a linear cosine transform of a log power spectrum on a nonlinear mel scale frequency.

MFCCs, however, are not limited exclusively to speech problems. They are nowadays frequently employed in tasks such as genre classification in music information retrieval (Muller (2007)) and music similarity measures (Jensen *et al.* (2006)).In Biatov *et al.* (2008) MFFCs are used jointly with energy and delta coefficients as features in retrieval experiments involving 15 environmental audio events such as: airplanes, applause, car motors, car accidents, bar/restaurants, laughter, traffic, car races, town, casino, horses, weather, steps, crowds and explosions. In the already cited work of Lu & Hanjalic (2009), MFCCs are used in conjunction with other spectral features (sub-band energy ratios, brightness, bandwidth) and temporal features (short time energy, zero crossing rate), for describing the content of a composite audio stream, thus including a set of sounds of various nature.

Throughout all our experiments, then, the parametrization of the audio signal, whatever the task and the audio source, is achieved by using MFCCs and the energy coefficient for each frame of signal. First and second order derivative features were not used as in apposite experiments they were shown to bring negligible improvement in the recognition capability of the system, while significantly slowing down critical parts of the computation.

3.5 Similarity detection and score

The similarity detection and score subtasks amount to identifying and quantifying the similarity of portions of audio segments. More explicitly, in our context, it consists in determining whether a repetition of a query occurs somewhere in its recent past. The library search aspect will be described later on but it can be anticipated that the problem can be reduced to that of searching in the past.

If the concept of distance is more straightforward when comparing strings or sequences of symbols, the idea of similarity or distance is less intuitive for acoustic sequences (at least from the point of a view of a machine). This stems from the fact that different occurrences of strings are exactly identical, and confusion matrices might be used to define distances between the elementary components of a string (symbols). Moreover, segmentation of strings into symbols is straightforward. Segmentation of a continuous audio stream into elementary units (phones, words, phrases) is a more challenging task; in addition, different instances of a word exhibit great variation that

make more difficult the task of recognizing their common belonging to same linguistic class.

Consider the following example, detailed by Fig 3.3. The spectrogram of three spoken sentences (see caption to know about the text) are reported. The similarity detection and score subtasks have the primary end goal of recognizing that:

- the first and second sentence shares a word in common, the word being *ambassadeurs*.
- the first two sentences do not share any word in common with the third one (at least disregarding article and prepositions, which are anyway too short to even meet the minimum length condition of our targeted motifs).

The objective is to recognize detections and their exact location within the utterances analyzed, at the signal level. This has to be accomplished in the absence of:

1. any prior knowledge on the presence of repetitions within the acoustic segments.
2. any labeled occurrence of speech units like words and phones.

One could think of reducing the problem to a string matching one, by properly mapping a sequence of acoustical features into a sequence of symbols. But that would make the recognition performance dependent on the transformation function, and likely require some training data to find the proper mapping function.

Given the assumptions of our framework, we have instead resorted to the scoring of the alignment of speech sequences. The technique that we will present in the following is the well-known dynamic time warping, extensively used for aligning speech sequences to account for differences in speaking rate and to provide a (dis)similarity score of those patterns.

3.5.1 Dynamic Time Warping

Dynamic time warping (DTW) is a well-known template matching technique for aligning two sequences in an optimal sense (according to some metric) and for scoring their similarity. It consists in finding the best mapping between them by warping one or both and by using dynamic programming (DP) relations.

Besides speech processing, it has been successfully applied to a number of different domains and applicative contexts, like data mining Keogh & Pazzani (2000) and hand writing recognition Munich & Perona (1999).

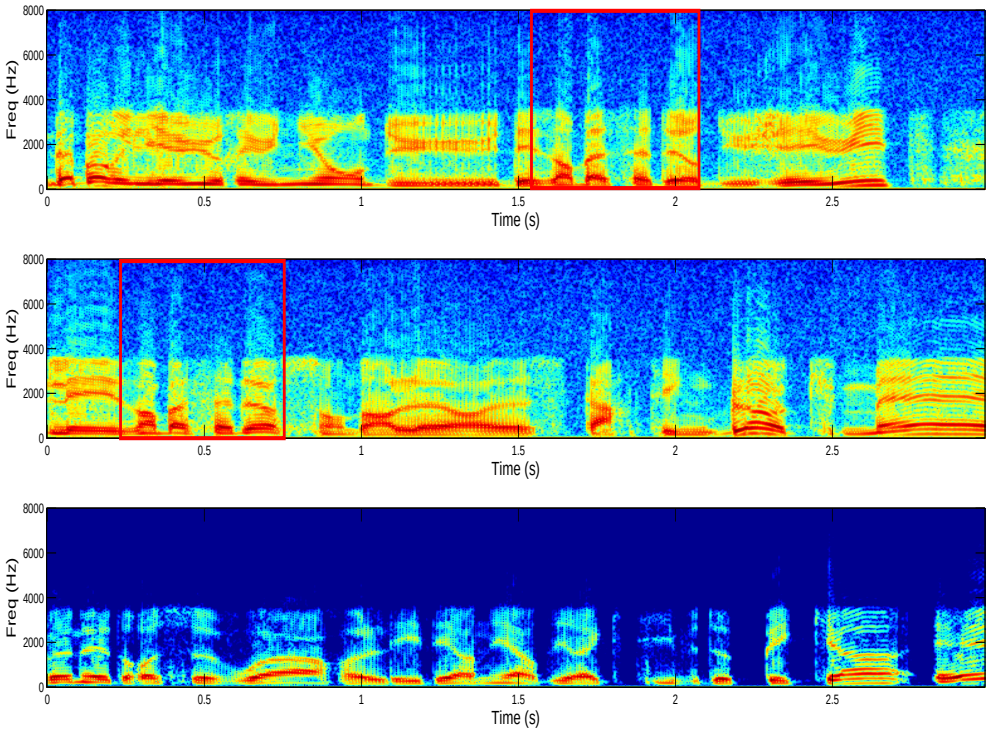

Figure 3.3: Spectrogram for the expressions (from top to bottom): *d'abord il y a eu une reunion des ambassadeurs du G8*; *oui, les ambassadeurs sont dans le starting block, mais*; *il venait de recevoir les derniers directives de Pekin.* The red rectangles mark occurrences of the word *ambassadeurs*.

Formally, let consider two sequences of vectors:

$$U = u_1, u_2, \ldots, u_M \tag{3.4}$$

$$V = v_1, v_2, \ldots, v_N \tag{3.5}$$

A warping path P is a set of pairs (i, j) that defines a mapping between U and V while satisfying a set of constraints. Formally:

$$P = \{(i_1, j_1), \ldots, (i_{L(P)}, j_{L(P)})\} = \{(i_k, j_k)\}_{k=1}^{L(P)} \qquad \max(M, N) \leq L(P) < M+N-1 \tag{3.6}$$

A warping path is characterized by

- a length $L(P)$, which is the number of path entries.
- a cumulated distortion $D(P) = \sum_{k=1}^{L(P)} d(i_k, j_k)$, where d measures the distance between two vectors, according to some metric (Euclidean, Mahalanobis, cosine, etc.).
- an average cumulated distortion, or normalized path weight $W(P) = D(P)/L(P)$.

The conditions that a warping path is required to fulfill are the following:

- **Boundary conditions**: $P_1 = (1, 1)$ and $P_L = (M, N)$. That means each path starts and ends at diagonally opposite corner cells of the matrix. The boundary constraint results from the assumption that the endpoints of the speech patterns are given a priori, after some speech-detection operation (Rabiner & Juang (1993)).
- **Continuity**: Given $P_k = (i_k, j_k)$ and $P_{k-1} = (i_{k-1}, j_{k-1})$, then $i_k - i_{k-1} \leq 1$ and $j_k - j_{k-1} \leq 1$. This restricts the admissible steps in the warping path to adjacent cells. The continuity requirement is enforced in order to prevent the loss of any information in the time alignment.
- **Monotonicity**: Given $P_k = (i_k, j_k)$ then $P_{k-1} = (i_{k-1}, j_{k-1})$ where $i_k - i_{k-1} \geq 0$ and $j_k - j_{k-1} \geq 0$. This requires all the path entries to be monotonically spaced in time. Monotonicity is needed to preserve the temporal order of the spectral sequence, which is crucial for the *linguistic meaning* of time normalization.

In fact, a large number of constraint types have been proposed in speech recognition to specify the path properties (local continuity constraints, global path constraints, slope weightings) to model speaking rates and temporal variations in speech utterances; a comprehensive tutorial can be found in Rabiner & Juang (1993).

The main questions in time-aligning two speech patterns and deeming their similarity are:

1. how to score a given path?
2. which path is to be chosen for providing a unique dissimilarity measure?
3. how to efficiently compute this path?

Path scoring. A global pattern dissimilarity measure for a given path can be defined as:

$$d_P(U,V) = \sum_{k=1}^{L(P)} d(i_k, j_k) m(k) / \Phi_P \tag{3.7}$$

where d the local distance previously defined, $m(k)$ is a nonnegative weighting coefficient and Φ_P is a normalizing factor, defined as $\Phi_P = \sum_{k=1}^{L(P)} m(k)$, which serves to have an average path distortion independent of the length of the patterns being compared.

Best path definition. Among the admissible paths, the natural choice for the best path is the one that minimizes this measure:

$$\hat{P} = \arg\min_P d_P(U,V) \tag{3.8}$$

and the respective score is the dissimilarity measure of the two patterns. This choice has an obvious meaning if we think of two utterances of a same word being compared: the dissimilarity is measured on the best path since is the one that accounts for nonlinear differences in speaking rate between the two occurrences of the same word.

Finding the best path through DTW. Dynamic Time Warping provides a solution to efficiently compute the minimization in (3.8), based on the use of dynamic programming (Bellman (1957)), which is a tool for solving sequential decision problems. Following the local optimality principle, expression 3.8 can be solved by noting that:

if $C = \{(1,1)\cdots(i,j)\}$ *is the optimal path joining* $(1,1)$ *to* (i,j)*, then for any* $(i',j') \in C$*, the optimal path leading from* (i',j') *to* (i,j) *is included in* C*.*
Then, the accumulated distortion at each cell (i,j) can be computed as:

$$D(i,j) = \underset{(i',j')\in V(i,j)}{\arg\min}\ D(i',j') + d_W((i',j'),(i,j)) \tag{3.9}$$

where (i',j') belongs to the neighborhood of (i,j) defined by the local constraints, and d_w is a weighted $d(i,j)$ according to the local path from (i',j') to (i,j).

3.6 Summary

In this chapter we have first formally defined the task at stake, then proposed to approach motif discovery in a modular fashion, according to a division into elementary subtasks. Based on this paradigm, we have defined each subtask, proposing appropriate solutions, drawing from well-known state of the art techniques. In the next chapter we will see how these subtasks can be integrated together to design a (preliminary) discovery system. In particular, we will show how appropriate variations of DTW can be proficiently used to perform the similarity detection and score subtask.

Chapter 4

Initial steps towards efficient motif discovery

This chapter focuses on our initial approach to motif discovery, detailing various options for the similarity detection and score subtasks. More specifically, three variations of DTW are described that enable partial sequence alignment of speech sequences, namely a) segmental locally normalized dynamic time warping (SLNDTW) b) band relaxed SLNDTW and c) fragmental SLNDTW.

The need for subsequence alignment techniques originates by the unknown endpoints locations of word units within the continuous audio stream. Therefore, the proposed modifications accomplish two main tasks: a) they identify the endpoints of likely repetitions, and b) provide a dissimilarity score to qualify them or not as motif occurrences. Besides, the integration of each of these procedures into the discovery architecture is described, before concluding by highlighting the limit of the initial system proposed. This will be instrumental in introducing the more definitive architecture based on seeded discovery in the next chapter.

While part of the similarity detection and score subtasks, the discussion on the library search aspect will be postponed to the subsequent chapter, since it presents specificities that deserves a special attention. It can be anticipated, however, that the pattern matching techniques employed in library search will be the same as the ones introduced in this chapter.

4.1 Dealing with unknown word endpoints: the need for local alignments

The boundary constraints of global alignment methods result from the assumption that the endpoints of speech units are well known a priori. Whether these units are phones, words, phrases, once the information about their exact endpoints is available, the direct comparison by global alignment and path scoring can be performed, to decide if they are similar or not. This is what DTW accomplishes, in the classical version detailed in the previous chapter.

However, endpoints location of speech units is a source of knowledge that is precluded to us, according to the basic assumptions of the motif discovery problem. Therefore, repeating patterns have to be automatically extracted from the unsegmented stream. Referring to the ARGOS framework, the problem translates into that of finding repetitions of the query in its search buffer or in a motif in the library. To this end, we ask how to adapt DTW so as to exploit its attractive capability of quantifying similarity of audio sequences while removing the limiting assumption of knowing patterns' endpoints. We provide an answer to this question, and propose a solution for similarity detection, that consists in enabling *local alignment* by relaxing the canonical boundary constraint, and by properly normalizing local alignment paths. The strategy basically works by identifying some *likely* matching pairs of segments, and by using dynamic programming relations and path distortion measures to compute and score the corresponding alignment path.

Optimality principle for boundary relaxed alignment. To enable the computation of local alignments while still resorting to dynamic programming, the optimality principle must be redefined to account for the relaxation of the boundary constraints.

In (di Martino (1985)), it was proposed a generalisation of the optimality principle to account for the relaxation of the boundary condition, when endpoints detection in speech is corrupted by noisy environment. It can be restated as:
If $C = \{s(i,j), \ldots, (i,j)\}$ *is the optimal path starting from* $s(i,j)$ *and reaching* (i,j)*, then for any* $(i',j') \in C$*,* $s(i',j') = s(i,j)$ *and the optimal path from* $s(i',j')$ *to* (i',j') *is included in* C*.*

According to this new formulation, to evaluate recursively the cumulated distortion $D(i,j)$, the different lengths of the warping paths ending in (i,j) are to be

considered and properly normalized; this is slightly different from DTW, where local path computation is influenced only by the local distance d of the entry (i, j) and by the cumulated distortion D of the partial paths merging into (i, j). The following set of equations (*local normalization*) is obtained by accounting for length normalization; they explicitely tell how the local path, the distortion, the starting point and the length of the path passing to (i, j) are obtained. By indicating with $(\hat{i}', \hat{j}')$ the winning entry of the neighbourhood of (i, j):

$$(\hat{i}', \hat{j}') = \underset{(i',j') \in V(i,j)}{\arg\min} \frac{D(i', j') + d_W((i', j'), (i, j))}{L(\{s(i', j') \dots (i', j')\}) + L(\{(i', j') \dots (i, j)\})} \tag{4.1}$$

$$D(i, j) = D(\hat{i}', \hat{j}') + d_W((\hat{i}', \hat{j}'), (i, j)) \tag{4.2}$$

$$s(i, j) = s(\hat{i}', \hat{j}') \tag{4.3}$$

$$L(\{s(i, j) \dots (i, j)\}) = L(\{s(\hat{i}', \hat{j}') \dots (\hat{i}', \hat{j}')\}) + L(\{(\hat{i}', \hat{j}'), (i, j)\}) \tag{4.4}$$

Building from this new set of relations, we can then start illustrating our solution to enable *partial* sequence matching by dynamic programming.

4.2 Segmental locally normalized DTW

The end goal of local alignment is to find two matching subsequences $u_{i_s} \dots u_{i_e}$ of a query U and $v_{j_s} \dots v_{j_e}$ of the search buffer V, and the respective *matching path* $\hat{P} = \{(i_s, j_s), \dots, (i_e, j_e)\}$ with $1 \leq i_s \leq i_e \leq M, 1 \leq j_s \leq j_e \leq N$. More specifically, we define a path P as *matching* when its score $W(P) < \epsilon$, being ϵ a proper similarity threshold (it plays indeed the role of the threshold in the relation 3.1).

It is first considered the case $i_s = 1$ and $i_e = M$, that is a whole repetition of the query is searched in the search buffer. This is achieved by:

1. Identifying the starting point of *likely* matching paths among the $(1, j)$ entries.
2. Computing each path, using the set of local normalization relations introduced in the previous section.
3. Reconstructing the matching paths, if any, by backtracking from the respective ending points.

The algorithm is termed segmental locally normalized dynamic time warping (SLNDTW), since:

- it enables the alignment of the query with multiple subsegments of the search buffer (hence segmental).
- it performs path computation by local normalization (hence locally normalized).

4.2.1 Algorithmic description

As a potential match can occur anywhere in v, a strategy is needed that allows $j_s \neq 1, j_e \neq N$.

Starting point selection. The starting point of a path is determined by hypothesizing the presence of the start of a matching path, whenever a *sufficiently* small value of local distance $d(1,j)$ is observed. The value of $d(1,j)$ is compared with the average weight W of $P = \{(1,j_s),\ldots,(1,j-1),(1,j)\}$, where $(1,j_s)$ is a generic starting point previously selected. This path results from the addition of $(1,j)$ to the already existing path passing through its left neighbour. The underlying assumption is that if $d(1,j)$ is smaller, then a matching path is more likely to start from $(1,j)$ than $P = \{(1,j_s),\ldots,(1,j-1),(1,j)\}$ being a matching path itself.

The procedure is formally described by the following expression: $\forall j, 1 \leq j \leq N$,

$$\begin{cases} D(1,j) = d(1,j) \\ L(1,j) = 1 \end{cases} \text{, if } d(1,j) < \frac{D(1,j-1)+d(1,j)}{L(1,j-1)+1}$$
$$\begin{cases} D(1,j) = D(1,j-1)+d(1,j) \\ L(1,j) = L(1,j-1)+1 \end{cases} \text{, otherwise} \tag{4.5}$$

Path computation. Except for $i = 1$, each path is computed by iteratively applying the local normalization recursion, which consists in minimizing, at each point (i,j) of the computational grid $[1,\cdots,M] \times [1,\cdots,N]$, the weight $W(i,j)$, that is the quotient between the accumulated distance $D(i,j)$ and the path length $L(i,j)$. Formally:

$$W(i,j) = \min\left[\frac{d(i,j)+D(i-1,j)}{L(i-1,j)+1}, \frac{d(i,j)+D(i-1,j-1)}{L(i-1,j-1)+1}, \frac{d(i,j)+D(i,j-1)}{L(i,j-1)+1}\right] \tag{4.6}$$

Note that relation (4.6) is formally identical to (4.1), when the neighbourhood of cell (i,j) is composed of the adjacent cells $(i-1,j),(i-1,j-1),(i,j-1)$ and no weighting slope is applied [1].

[1] local conditions and path constraints will be properly specified in the experiments

Match identification After path computation, the weight of all computed paths is stored in the entries $W(M, j), 1 \leq j \leq N$.

Every subsequence $v_{j_s}, \ldots, v_{j_e}$ for which a path $P = \{(1, j_s), \ldots, (M, j_e)\}$ exists such that $W(P) < \epsilon$, is a repetition of the query U. Note that, since speech is a monotonous signal, there are no big changes between adjacent distances, so the cells (M, j) close to a matching ending point (M, j_e) might also be matching ending points, although related to strongly overlapping subsequences of $v_{j_s}, \ldots, v_{j_e}$. We consider then the minimum weighted path as the privileged matching path, assuming we are interested in only one possible repetition (we will see how this will be indeed the case within the motif discovery system). In alternative, all possible matching subsequences can be retained, as long as they do not significantly overlap in time (in this case, they would refer indeed to the same repetition).

4.2.2 Example Output

A specific example of the application of SLNDTW is shown in Fig. 4.1, where an instance of the word *ambassadeurs* is searched within the phrase *d'abord il y a eu une reunion des ambassadeurs du G8*, spoken by the same male speaker. The top half of the picture shows the spectrograms of the two utterances and the respective distance matrix.

After SLNDTW computation, the entry $(M, j), 1 \leq j \leq N$ that minimizes the average path distortion $W(M, j)$ is selected as the ending point of the best path (as can be observed in the bottom half of the figure). The continuous red line superimposed to the low distortion diagonal region, indicates the successful reconstruction of the alignment between the two occurrences of *ambassadeurs*.

Moreover, is interesting to observe the distortion profile of the matching path, that is the sequence $\{d(i_k, j_k)\}_{k=1}^{L(P)}$, depicted in Figure 4.2. Far from being a flat curve, the profile exhibits a very irregular pattern, showing several peaks and valleys.

4.2.3 Integrating SLNDTW in motif discovery

The integration of the technique in the motif discovery system is straightforward. In Fig 4.3 the search for a repetition of the query in its search buffer by SLNDTW is depicted. The paths satisfying the similarity condition ($W(P) < \epsilon$) are sorted in ascending order according to the dissimilarity score, and evaluated until a match is

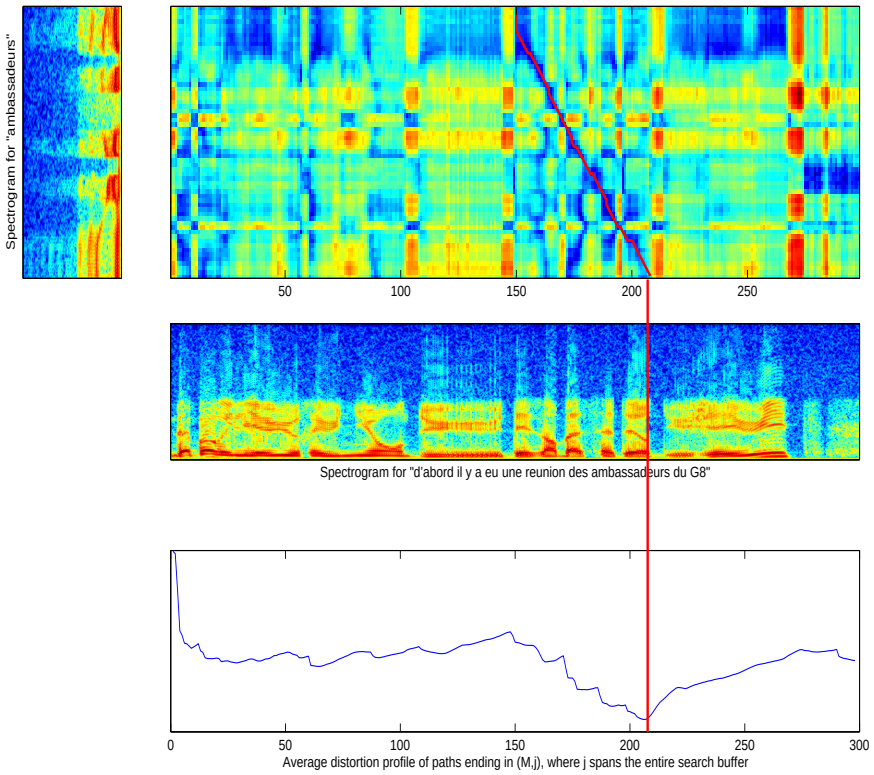

Figure 4.1: Application of SLNDTW to the retrieval of the word *ambassadeurs* within the phrase *d'abord, il y a eu une reunion des ambassadeurs du G8*. The procedure correctly identifies the two repetitions and tracks the corresponding alignment path from the ending point in the last row of the distance matrix. It can be observed as the ending matching point corresponds to the minimum of the average distortion among the paths ending in (M, j).

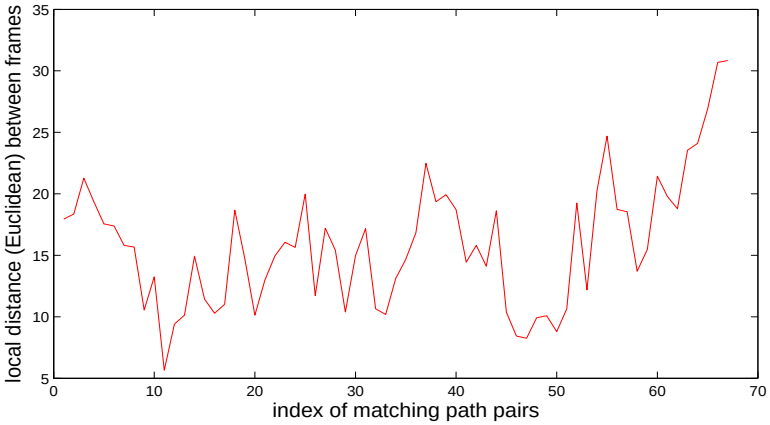

Figure 4.2: Distortion profile of the matching path (Euclidean distance used).

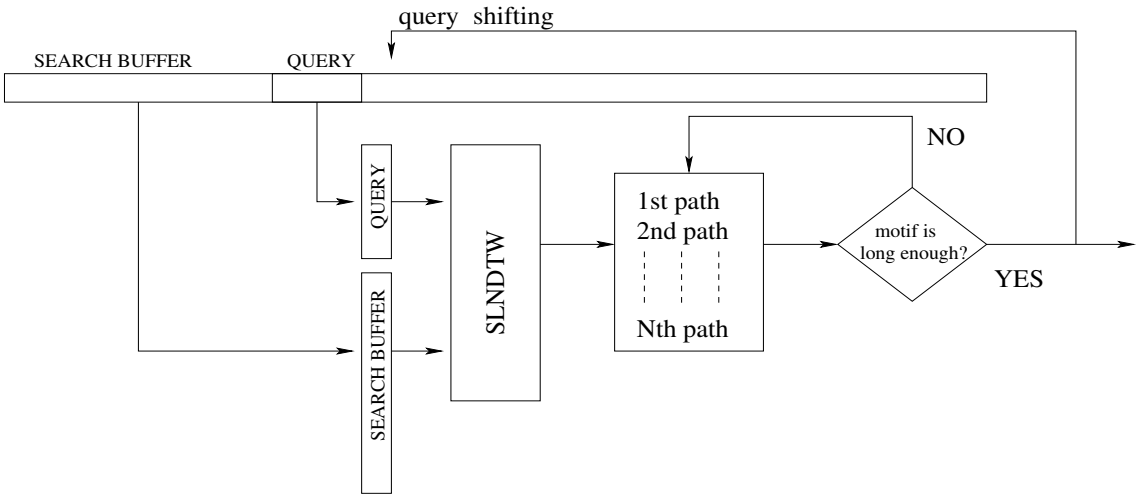

Figure 4.3: Integration of SLNDTW in the query-search buffer search for a repetition.

found. We call these paths (or equivalently, the subsequences they map) as *candidate motifs* (they are indicated by the N paths in Fig 4.3). The presence of a match is determined by evaluating the length of the two occurrences found; if they obey the minimum length requirement a motif is discovered. Actually, as one of the occurrences is the query itself, which must necessarily meet the minimum length constraint, the fulfillment of this condition is verified only for the occurrence in the search buffer. In fact, because of *spurious* mappings, a query might be mapped into a shorter

subsegment of the search buffer, possibly violating the length constraint.

One noteworthy observation is that, by retaining just one matching path, we collect, at most, one repetition of the query in its search buffer, regardless of the possible presence of multiple occurrences. This strategy comes from the assumption that two instances of a same motif cannot simultaneously occur in the same search buffer; they should have already been detected in a previous step of the algorithm, when the second occurrence was taken as a query; hence the current occurrence, the third one in temporal order, should have been already identified as a repetition by a previous library search. However, missed detection of occurrences might always occur, because of possible deficiencies in pattern comparison or in the segmentation strategy (we will discuss next an example of those); therefore, a solution might be convenient that envisions the possibility of detecting several repetitions of the currenty query.

The integration of SLNDTW in the ARGOS framework rises several issues, mainly concerning the choice of the query length and the selection of subsequent queries.

The choice of the query length. The framework described is satisfying only if all motifs and motif occurrences are equally long, as the query length is fixed. The only variability admitted is in the mappings performed by alignment, that can map a segment into another of different (but usually similar) length. Allowing only the detection of repetitions of a fixed length query, is a very limiting requirement for a system supposed to be applicable to a variety of discovery tasks and motifs.

One possible solution is to consider the motif length as a user specified parameter (see Lin *et al.* (2002) and Minnen *et al.* (2007)). This is impractical because implies the algorithm to be run several times while varying the query length. Moreover, giving this information as input clearly contrasts with the unsupervised learning paradigm.

Extraction of a subsequent query. A subsequent query can be chosen so as to partially overlap with the preceding one, to deal with the possibility of a motif occurrence in between. If the current query is defined by time endpoints $[t_0, t_0 + \Delta]$, the next query extends from $t_0 + \Delta - L_{\text{overlap}}$ to $t_0 + 2\Delta - L_{\text{overlap}}$, where L_{overlap} quantifies the degree of overlap between adjacent queries. However, the only optimal choice to guarantee that a repetition does not get stuck in between two queries is $L_{\text{overlap}} = L_{\text{query}} - 1$ (of course under the assumption that a motif occurrence is exactly L_{query} long). That means each query is just shifted of one frame and then searched. Such solution implies an explosion of the number of queries and search operations.

Alternative values of L_{overlap} would speed up the algorithm at the cost of being suboptimal. In fact, even slight misshifts between a query and a motif occurrence can result in a high distortion alignment path that prevents the detection of a repetition. This is in fact an example of a deficiency in pattern matching (but also in segmentation strategy) that can lead to miss repetitions.

We will see in the following alternative solutions to (partially) overcome this problem.

4.3 Band Relaxed SLNDTW

SLNDTW aims at finding repetitions of the query in the search buffer. This approach would be effective only if motifs were of fixed length and exactly coincident with the query extracted from the stream. But in practical scenarios that would strongly limit the successful application of the system. For example, one might wonder how to handle the discovery of jingles of a few seconds and songs of several minutes within the same framework, given the different duration of those sound patterns.

If the assumption on the length restricts the number of retrievable motifs, time synchronization mismatch between a repeating segment and the query impacts the recognition capabilities of the algorithm. Since the endpoints of sound patterns are not known, a pattern recognition technique that strongly depends on the synchronization of query and motif, is clearly unsatisfying.

The idea is to generalize the segmental property of SLNDTW to relax boundary constraints also on the query. We propose a variation of SLNDTW called band relaxed SLNDTW that relaxes the constraints that forces a path to start in $(1, j_s)$ and end in (M, j_e). This is accomplished by permitting the selection of starting and ending points within a group of rows (a band, indeed) in the distance matrix, instead of a single row, while constraining the paths to cross a central band. This heuristic allows the retrieval of variable length motifs, as possible matches can be as short as the central band, or as long as the entire query length. Besides admitting a certain variability in the length of repetitions, it also partially mitigates synchronization issues between a motif and a query, since they are required not to coincide but at least to match in the central band.

We detail in the following the basic idea and the practical implementation of the procedure.

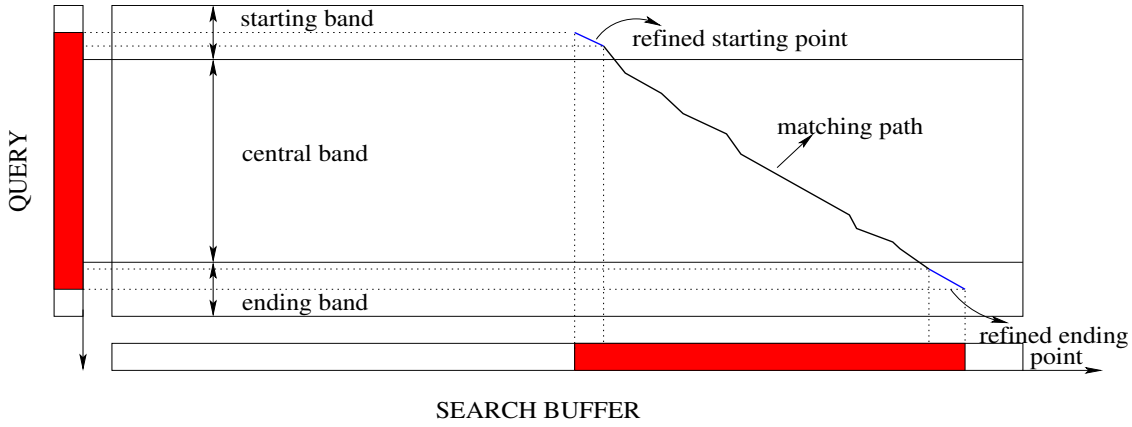

Figure 4.4: Band relaxed SLNDTW: the motif completely includes the central band. After path reconstruction, boundaries are refined in the starting and ending band (blue lines).

4.3.1 Algorithmic description

The computational lattice $[1, \cdots, M] \times [1, \cdots, N]$ is divided in three horizontal bands:

- the starting band, including the entries $(i, j)|i \in [1, L_s]$. Each entry in this band is a potential starting point of an alignment path.
- the central band, including the entries $(i, j)|i \in]L_s, L_s + L_c]$ that all paths are constrained to cross.
- the ending band, where paths end, includes all points $(i, j)|i \in]L_s + L_c, M]$. Each entry of the ending band is a potential ending point of an alignment path.

Since paths are forced to pass through the central band, and can start and end anywhere in the starting and ending band, the length of the segments mapped can vary in the range $[L_c, M]$.

The algorithm comprises the following steps:

1. $\forall (i, j)|i \in [1, L_s]$:

   if $d(i,j) < \left[\frac{d(i,j) + D(i-1,j)}{L(i-1,j)+1}, \frac{d(i,j) + D(i-1,j-1)}{L(i-1,j-1)+1}, \frac{d(i,j) + D(i,j-1)}{L(i,j-1)+1} \right]$

   then (i, j) is the starting point of a new path, otherwise it is added to the path that minimizes $W(i, j)$. Note that this is a generalization of equation (4.5),

as the same condition is expressed by considering the whole neighbourhood of (i, j) rather than the single cell at its left $(i, j-1)$.

2. $\forall(i,j)|i \in]L_s, M]$ compute path as in (4.6).

3. $\forall(i,j)|i \in [L_c{+}L_s, M]$ select the ending point of a match, if any, as in SLNDTW, and reconstruct the corresponding path.

4.3.2 Path boundary refinement

In addition, a boundary refinement strategy is adopted that seeks to possibly extend the matching path by appending new pairs from the starting and ending points. The reason for such a strategy is twofold:

1. to mitigate the effect of possible imperfections in the endpoints detection.

   To better understand where these imperfections originate, consider how the ending points are selected. For each path, the corresponding ending point is identified as the minimum-valued point (a valley) of the path average distortion in the ending band. The underlying assumption is that the valley indicates the end of the match while other points, where the distortion increases, indicate a mapping between non-matching frames of signal, hence discarded. This assumption not always holds true, as the distortion profile of a matching path can be quite irregular and can exhibit a variety of peaks and valleys, as can be observed from Fig 4.2. Therefore, while left out from the matching path, those additional points might effectively be part of the true match. Similar observations can be drawn for the starting point selection heuristic.

2. To favour the detection of sufficiently long matches. Depending on the specific parameter setting, the size of the central band might be shorter than $L_{\min}$. Without refinement by path extension, several matching paths were noted to be truncated (because of the imperfections described) generating matching segments too short to even be evaluated as motif occurrences. In this case, not only the endpoints are not exactly retrieved, but the motifs themselves are erroneously skipped.

Practically, refinement by path extension carries over as long as the average weight of the extended path does not increase too much. The steps of the procedure are summarized in the following (concerning the forward extension from the ending point):

1. Consider the path P with $W(P) = W_o$ ending in (i_e, j_e).
2. Select in the neighbourhood of (i_e, j_e) (composed of $(i_e{+}1, j_e{+}1), (i_e{+}1, j_e), (i_e, j_e{+}1)$) the point that, added to P, minimizes $W(P)$, and add it to P as its new ending point.
3. If $W(P) < W_o + kW_o$, then repeat the procedure from 1, otherwise remove the new ending point from P and stop the procedure.
4. Compute the new averaged weight W of the extendend path.

The same procedure applies when extending the path backward from its starting point (i_s, j_s). The term $W_o + kW_o$ is an adaptive similarity threshold used in place of the spectral threshold ϵ. It is used to limit the addition of *garbage* (that is, non matching frames of signal) to the matching segments; indeed, in case $W_o << \epsilon$, a significant number of high distortion d (non matching frames) would be added if $W(P) < \epsilon$ was to be used as a stopping condition. In practical experiments, the value of k has been set to 0.1 or 0.2. Using a threshold slightly greater than W_o ensures that the extended path yields a distortion profile similar to the original matching path, while allowing for a certain margin to compensate for local distortion peaks that might stop prematurely the extension.

4.3.3 Integrating Band Relaxed SLNDTW in motif discovery

The integration of band relaxed SLNDTW in the motif discovery system is straightforward and follows observations made in paragraph 4.2.3.

We insist in emphasizing that the main issues implied by the use of SLNDTW, are not entirely solved by this modified technique. Band relaxed alleviates the impact of time mis-shifting between a query and a repeating segment, and allows a certain variability in motif length. However, repetitions are forced to occur in the central part of a query, to permit the correct computation and score of the matching path. In general, imposing a constraint on the relative position of a repeating pattern within a query, is arbitrary since no assumption can be made on the location of those patterns anywhere in the data stream.

Herley (2006) explicitly derives a condition for successful pattern matching that relates the length of a repetition and the length of the query (assuming that a repetition is always included in a query). One might wonder if a solution can be thought that avoids these arbitrary assumptions.

Furthermore, motifs are still hypothesized to be no longer than the query length, and the problem of occurrences stuck in between queries still stands.

4.4 Fragmental SLNDTW

Band relaxed SNLDTW does not constrain a motif to coincide with a query, but it still assumes the motif to be located in the central part of the query to completely include its central band.

A simple generalization of the previous algorithms, called fragmental SLNDTW, allows the retrieval of a match regardless of its position in the query and opens the door for seeded discovery. The term fragmental is adopted as detection is accomplished by first retrieving a portion of a repeating segment, *e.g.* a fragment.

4.4.1 Algorithmic description

SLNDTW detects a match whenever a query coincide with a motif. By using queries sufficiently small to be included in a repeating pattern, at least one motif fragment is guaranteed to coincide with a query. Suppose to fix an upper bound $L_{\max}$ for a motif length. If $L_{\min} \leq L_{\text{motif}} \leq L_{\max}$, splitting a $L_{\max}$ long query into $L_{\min}/2$ long subqueries ensures that at least one fragment coincides with one of the subqueries. This fragment can be then retrieved by conventional SLNDTW; afterwards, the entire match can be recovered by path extension as in the boundary refinement stage of the band relaxed SLNDTW algorithm.

Formally:

1. Partition the grid $[1, \cdots, M] \times [1, \cdots, N]$ in horizontal bands of vertical length $L_{\min}/2$, such as the i-th band includes all point $(i, j) | (i-1) \cdot L_{\min}/2 + 1 \leq i \leq i \cdot L_{\min}/2$.

2. Perform a conventional SNLDTW in each band and reconstruct the matching path, if any.

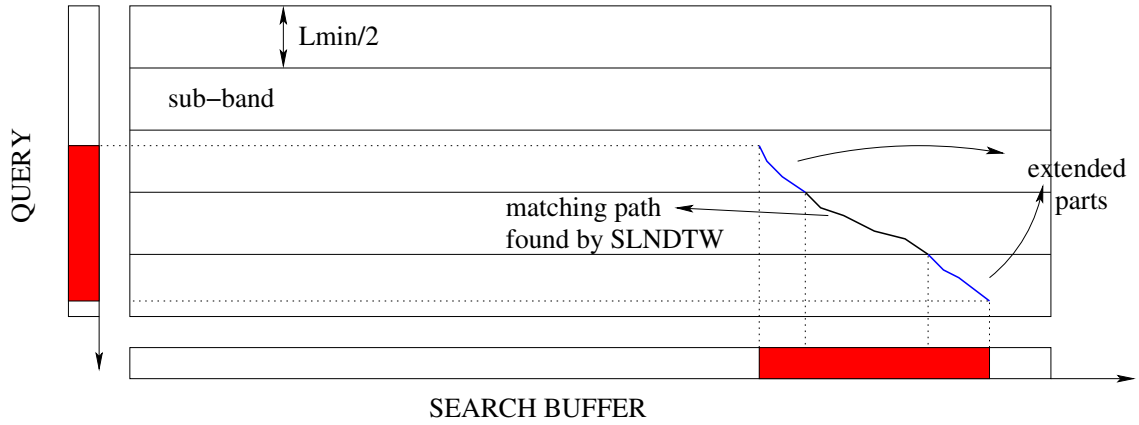

Figure 4.5: Fragmental SLNDTW: partitiong the query in $L_{\min}/2$ long subqueries ensures that a least a fragment of the motif coincide with a subquery. The entire match can be then recovered by extending the fragmental match.

3. Extend the matching path as in the boundary refinement strategy in Band SLNDTW.

This implementation has the advantage of enabling the retrieval of a match whichever its position in the considered query, hence it shows higher flexibility than the previous variations of DTW proposed.

Similar in principle is the DTW-based algorithm for partial sequence matching proposed in (Anguera *et al.* (2010)): while the starting point selection strategy and local constraints are slightly different, matching paths are similarly retrieved by local normalization and a subsequent path extension technique.

4.4.2 Integrating fragmental SLNDTW in motif discovery

Fragmental SLNDTW exploits the conditions on motif minimum and maximum length to recognize repetitions of a query subsegment in a part of an arbitrarily long search buffer. With respect to the previous methods, it does not constrain the relative location of the targeted repetition within the query.

Integrating these pattern matching techniques in ARGOS, two important issues were observed concerning:

1. the choice of the query length.

2. The position of subsequent queries, and the possible presence of repetitions in between.

While integrating fragmental SLNDTW in the discovery system, the following choices are made in relation to these aspects:

1. $L_{\text{query}} = L_{\max}$.
2. $L_{\text{overlap}} = L_{\max}/2$.

By operating these choices, it is guaranteed that each repetition, wherever occurring, is completely included in one of the queries extracted from the stream. And, as noted already, fragmental SLNDTW does not impose any condition on the specific location of a motif within a query, but only its inclusion in the query itself.

However, this framework still does not prove completely satisfying if one thinks of the possible implications of its applicability in motif discovery tasks. We list possible issues arising from this approach:

1. imposing a motif maximum length might be seen as a violation of the unsupervised learning paradigm. Besides a minimum length condition, which is reasonable to assume, one might want the algorithm to learn itself about any motif, of any possible length.

2. The 50% overlap between queries leads to inefficient computation since half the distance matrix between a query and the search buffer is computed a second time, when the query is shifted; and, even if library search has not been yet described, it easy to imagine that such information is computed a second time for each of the motifs in the library also. One obvious solution consists in just storing this information for reuse in the subsequent step, computing only the information coming from the novel portion of query (the right half of the segment). However, this could well turn impractical because of memory occupation issues: the quantity of information to be stored grows as new motifs are discovered, and the strategy could reveal unfeasibile even for small-sized libraries.

A solution is needed where neither a maximum length is imposed nor the use of overlap is needed, while ensuring for any possible repetitions to be detected. We will see an example of such a solution in the next chapter, where seeded discovery is presented.

4.5 Summary

In this chapter the initial step towards an efficient motif discovery architecture has been presented. This was attained by incorporating in the system three variations of DTW, with the aim of allowing partial sequence alignment of speech sequences. The limit implied by the use of these techniques has also been highlighted, consisting in too stringent constraints on the motif location, on their maximum admissible length, and in the use of overlapping queries. We will see in the next chapter how a straightforward modification of this system can naturally lead to overcome the aforementioned problems.

Chapter 5

Seeded motif discovery

In the two previous chapters, the different modules of an architecture for motif discovery were described. This architecture results from dealing with each subtask comprising the global motif discovery task. However, some undesirable properties were noted in the pattern recognition techniques proposed to detect similarities in audio. In particular, two main issues arise from the use of the so called *fragmental SLNDTW* that push for further improvement:

1. The a priori estimate of a motif maximum length.
2. The need for overlapping queries to deal with the possibility of repetitions occurring at the intersection of adjacent queries.

Motivated by the necessity of removing these constraints, a slightly different strategy is illustrated that can be straightforwardly applied to various motif discovery tasks. Moreover, we elaborate on the library search aspect, before presenting an algorithmic view of the final system and its different modules.

5.1 Seeded discovery

The proposed strategy, that represents our definitive proposition for motif discovery, is based on a matching technique that results from a slight variation of fragmental DTW. The end goal is to overcome the limits shown by the initial framework. The illustration of the system will be performed by presenting the underlying idea, the algorithmic implementation of the technique and its incorporation in the global architecture. The

system is named *seeded* discovery; the reason for such designation will be clear after its description.

5.1.1 From fragmental SLNDTW to seeded discovery

Fragmental SLNDTW was introduced to enable the detection of matches, independent of their relative position within a query. However, it was still assumed that each repetition is entirely included in the current query. This condition implied the need for overlapping queries to prevent the possibility of motif occurrences stuck between subsequent queries.

From fragmental SLNDTW... In fragmental SLNDTW, path computation in each sub-band is performed independently; it consists in performing a conventional SLNDTW, followed by a path extension stage from the endpoints of the best path selected. In path extension, the local distances in the neighborhood of the endpoints (belonging to different sub-bands), are evaluated to compute the local path according to the dynamic programming recursion. The extension is limited by two factors:

1. the stopping condition on the average distortion of the extended path.
2. the boundaries of the query, beyond which the path cannot be further lengthened. Therefore, matching segments stretching outside a query's endpoints are not retrievable in their entire duration (this is why, to guarantee the total inclusion of a repetition in the query, the $L_{\max}$ query length and the 50% overlap condition are used).

...to seeded discovery. The simple trick to overcome the boundary limitation consists in just removing this constraint to allow for the recovery of the whole match, whatever its length. Each of the $L_{\min}/2$ long subqueries is considered as an independent query to search for. If a match is found by SLNDTW, the extension is performed without any boundary constraint, reconstructing the motif occurrences in their entire length.
The discovery process, thus, amounts to identifying two (potential) fragments of two motif occurrences and their subsequent, unbounded extension: we call the potential motif fragment searched in its past as the *seed block* and the identification of a repetition by a matching path as a *seed match*. The term seed is metaphorically used,

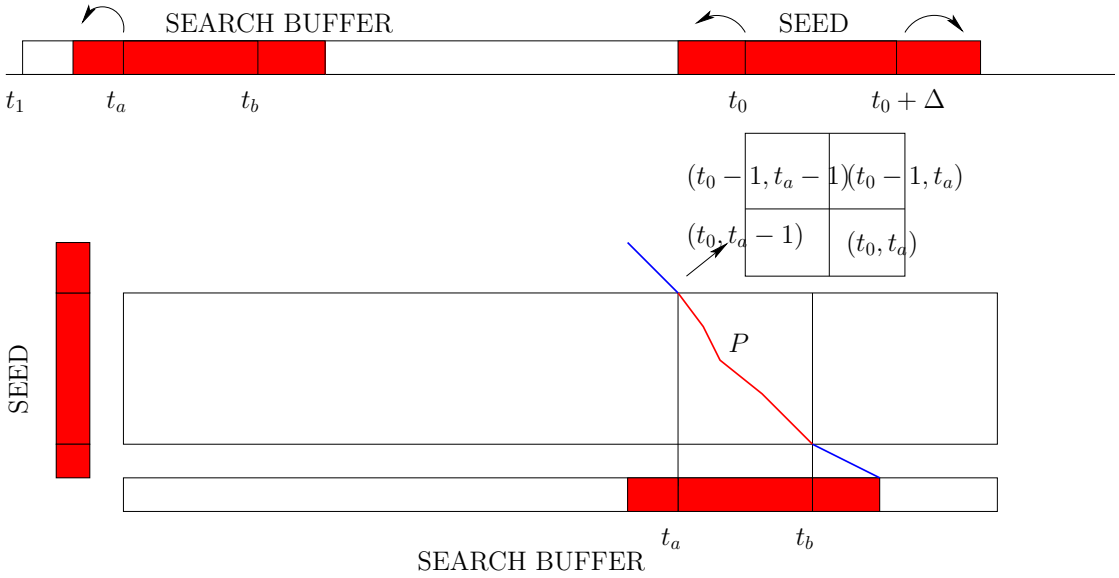

Figure 5.1: Seeded discovery.

as the seed block plays the role of an *embryonic* entity from which the final motif is grown. Since discovery is triggered by the detection of a seed match, the process is called *seeded discovery*.

A detailed description of its incorporation in the motif discovery framework is provided in the remainder of this chapter.

5.1.2 Algorithmic description

Consider a query of length $\Delta = L_{\min}/2$ defined by the endpoints $[t_0, t_0 + \Delta]$, and a search buffer defined by the endpoints $[t_1, t_0 - 1]$.

The search for a repetition is performed by computing a conventional SLNDTW between the two segments and reconstructing the best path P. From Figure 5.1, P identifies two matching segments, the query itself and the subsegment with endpoints $[t_a, t_b]$ in the search buffer. In order to retrieve the matching occurrences in their entire length, the path extension is performed by recovering the neighboring frames of the segments' endpoints directly from the stream.

In the following, the global framework will be illustrated: it will be explained how it implies the overcoming of the main issues noted for the initial system.

5.1.3 Integrating seeded discovery in motif discovery

Let us consider in the stream χ a time window $[t_0, t_0+\Delta]$ with $\Delta = \frac{L_{\min}}{2}$ and its past $[t_1, t_0 - 1]$. Indeed $\chi_{t_0+\Delta}^{t_0}$ is the seed.

A seed match is found if there exists a segment χ_c^d with $t_1 < c < d < t_0$ such that $H(\chi_{t_0}^{t_0+\Delta}, \chi_c^d) < \epsilon$. In order to check for the existence of a motif, the seed match is extended using $[t_0, t_0 + \Delta]$ and $[c, d]$ as anchor points. If there exist the pairs $a' < t_0, b' > t_0 + \Delta$ and $c' < c, d' > d$ such that:

$$H(\chi_{a'}^{b'}, \chi_{c'}^{d'}) < \epsilon \tag{5.1}$$

$$H(\chi_{a''}^{b''}, \chi_{c''}^{d''}) > \epsilon, \quad \forall a'' < a', b'' > b', c'' < c', d'' > d' \tag{5.2}$$

$$|b' - a'| \geq L_{\min} \tag{5.3}$$

then a motif is found with occurrences $\chi_{a'}^{b'}$ and $\chi_{c'}^{d'}$.

Conditions (5.1) and (5.3) are the same as relations (3.1) and (3.3), applied to the pairs (a', b') and (c', d'). They imply that $\chi_{a'}^{b'}$ and $\chi_{c'}^{d'}$ are sufficiently similar and long to be motif occurrences.

Condition (5.2) states that the match cannot be further extended beyond $[a', b']$ and $[c', d']$ without infringing the similarity condition in (3.1).

Much like in the initial system, seed matches are sorted in ascending order according to the similarity score. Each seed match is analyzed and undergoes the extension procedure until a match is found, that is the lengths of the matching segments obey the minimum length condition.

Once a comparison is performed, the pair seed-search buffer is shifted appropriately along the stream, that is either the new query is adjacent to the repetition found (if any), or adjacent to the preceding query. Basically: if a motif was previously found with endpoints $[a', b']$, the new seed endpoints are $[b', b' + \Delta]$, otherwise $[t_0 + \Delta, t_0 + 2\Delta]$.

We remark the two main advantages of adopting this strategy:

1. there is no need for guaranteeing the inclusion of a repetition within a query and of setting an upper bound for a motif length. Since each query is $L_{\min}/2$ long, it is guaranteed that a repeating segment, whatever its length and position in the stream, includes a query as a fragment, thus retrievable by seeded discovery.

2. The use of overlapping queries is useless. After a motif is discovered, it is sufficient to consider as a query the $L_{\min}/2$ long segment adjacent to the repetition found.

This framework is straightforwardly applicable to a variety of discovery task. It does not constrain repetitions to occur in specific part of the stream, nor to have a limited length. There is no contraindication in using it for discovering words, jingles, songs (beside setting an appropriate similarity threshold or search buffer length, given the specific task). This will be clear when the result of the experimental evaluation on these different tasks will be reported in the subsequent chapters.

5.2 Library search

To complete the illustration of the motif discovery architecture, we turn to the the library search component of the system. The core of the problem revolves around two main aspects:

1. How to perform the comparison between a seed and a motif in the library.
2. How to represent (or *model*) a motif in the library.

The two aspects are strictly intertwined; in fact, the way a motif is represented strongly influences the choice of the pattern matching technique; vice versa, given the pattern matching technique, the representation of a motif is chosen to fit the mechanisms of the algorithm.

One natural solution is to let each motif be represented by the sequence of all its current occurrences, without any modelling. If N occurrences of a motif have been collected at a given point of the computation, then the current seed block is compared with all of them to determine whether a seed match has occurred or not. This comparison can be performed in different ways. We mainly focus on two common strategies:

1. generalize the pairwise dynamic programming alignments described to the alignment of N sequences. This turns impractical for more than a few sequences. In fact, this strategy implies the construction of an N-dimensional distance matrix, as well as the evaluation of $2^N - 1$ neighboring frames each time a local path is computed (both in SLNDTW and path extension).

2. Compare independently each seed with all N occurrences. This poses two problems, in turn: a) how to fuse the independent N scores and b)how to perform the path extension, since N different alignments have been produced.

While providing an answer to these questions is worthwhile, we have decided to apply to the search library problem, the seeded discovery described for the seed-search buffer search for a repetition. This is a case of choosing the pattern matching technique first, and adapting accordingly the motif representation. The reason for this choice is twofold:

1. it exploits the use of seeded discovery, which has proven successful in detecting repetitions in the seed-search buffer framework.

2. it forces the use of one template or *model* for a motif, which is computationally attractive, and straightforward in the application (in fact, it simply consists in searching a seed in a motif model, which plays the role of the search buffer).

Once the decision is made, the problem is that of appropriately modelling a motif from the set of its occurrences, so as to faithfully represent the underlying pattern.

Three main modelling strategies are discussed in the following:

1. average of occurrences

2. median of occurrences

3. random occurrence

5.2.1 Average of occurrences

Given two sequences of vectors $U = u_1, \ldots, u_M$ and $V = v_1, \ldots, v_N$ and their alignment path $P = \{(i_k, j_k)\}_{k=1}^{L}$, the average occurrence $A = a_i, \ldots a_L$ is defined as:

$$a_k = (u_{i_k} + v_{j_k})/2 \tag{5.4}$$

Accordingly, a model is built by simply averaging the contributions of each occurrence. Each time a new occurrence is collected, the model is updated by averaging the newly detected sequence. If a new occurrence V is averaged with a model representing N sequences, the model is weighted by a factor N in average computation, to account for the N occurrences; then the new model M is obtained from the old one A and V as:

$$m_k = (N * a_{i_k} + v_{j_k})/(N + 1) \tag{5.5}$$

This modelling strategy was already used in (Cheng *et al.* (2005)) to model speech sequences aligned by DTW.

5.2.2 Median occurrence

The median occurrence M of a set of N occurrences $m_i, i = 1, \ldots, N$ is defined as the occurrence closest to all the other ones, in average, according to a given dissimilarity score d:

$$M = m_i \text{ where } i = \arg\min_{1 \leq j \leq N} \sum_{k=1}^{N} d(m_j, m_k) \tag{5.6}$$

The reason for using this modelling strategy in place of the average is that median occurrence is less prone to degradation due to the detection of false hits. When this happens, the model resulting from averaging a false hit, is inevitably corrupted, in the sense that its representativeness of the underlying motif is decreased. On the other hand, in median modelling, if the new model is one of the true occurrences already collected, the quality of the model is not altered in any way by the collection of a false hit.

In the word discovery experiments presented in the next chapter an in-depth comparison of the performance of the two modelling strategies will be described.

5.2.3 Random occurrence

This type of modelling consists in randomly choosing one of the occurrences as representative of the underlying motif. This modelling strategy is used in discovery tasks where the targeted motifs are supposed to carry a limited variability, so that any occurrence is indeed extremely similar to all the other ones and can be assumed as representative. This is the model that will be used in the near duplicate discovery experiments, and the one employed in the original work (Herley (2006)), where the ARGOS segmentation was introduced.

5.3 Seeded discovery: algorithmic view and glossary of terms

The final architecture for motif discovery is depicted in Fig 5.2, where the single step of the iterative procedure is represented. Two main differences can be noted with respect to the previously proposed systems:

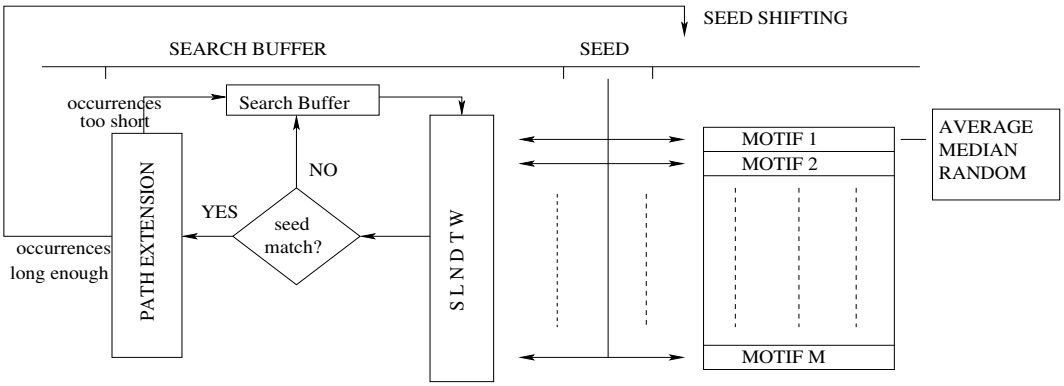

Figure 5.2: Motif discovery architecture based on seeded discovery.

1. the query is replaced by the seed.

2. the library search and the seed-search buffer comparison are explicitly shown to share the same pattern matching technique, based on SLNDTW + path extension.

Glossary of terms. For the sake of clarity, given the density of new concepts introduced in the last two chapters, we briefly summarize some important terms that will occur frequently in the remainder of the manuscript. Since some of these concepts are effectively similar and might generate ambiguities, we provide a more detailed explanation:

- **Match**: a match occurs whenever two segments are deemed as similar, according to the pattern matching techniques used and to the respective scores computed. This does not necessarily imply that they are also motif occurrences, as motifs are required to fulfill all conditions expressed in Section 3.1 (not just the similarity condition).

- **Matching path**: it is the alignment path mapping a pair of matching segments, as found by a DTW procedure.

- **Candidate motif**: in general, it might be used indifferently to indicate a match (hence, the pair of matching segments), or the respective matching path. With regard to the explicit seeded discovery framework, it indicates the N matching paths as computed by the SLNDTW algorithm (and the corresponding pair

of subsequences). The term *candidate* implies that these segments are to be validated as motifs in a subsequent stage. This stage might be represented by the sole path extension, or by further similarity score techniques. An example of these techniques will be seen when introducing the self-similarity comparison of speech templates in Chapter 6, and the downsampling of sequence in Chapter 7.

5.4 Summary

In this chapter we have introduced the concept of seeded discovery and we have illustrated its integration into a motif discovery architecture. In addition, the library search problem has been explicitly addressed, and a modelling of motifs is employed that permit to adopt the same pattern matching used in the search buffer comparison. The described architecture represents the definitive system that we propose for motif discovery tasks. In subsequent chapters we will see how additional modifications of this baseline framework can benefit its applicability to word and near-duplicate discovery tasks. We will describe in the next chapter the application of seeded discovery to word discovery experiments.

Chapter 6

Application to word discovery

In this chapter a first application of seeded discovery is presented in the form of word discovery in speech. We briefly define the task and underline the main challenges implied. The experimental evaluation is carried out on speech data by applying seeded discovery, first in the baseline form as described in the previous chapter, then in an improved version, aimed at dealing with variability issues. Conclusions are drawn both at quantitative and qualitative level to illustrate the capability of the algorithm to extract repeating patterns in speech data, as well as its limits.

6.1 Word discovery: definition and specificities

In this section, the task of word discovery in speech is defined and its peculiar aspects are described. We specify the type of motifs targeted, the different end goals with respect to ASR applications, and some typical properties, like the expected occurrence period of repetitions in speech.

The main challenge in discovering words is represented by the high variability of speech signal. In this regard, we enumerate the main factors responsible for such variability and their sources.

6.1.1 Definition of the task

The end goal of word discovery consists in identifying repetitions of acoustic patterns at the word level. The type of motifs targeted by this task is then a single word, or a short multi-word phrase, that repeats in spoken contents.

In human communications, word is the primary, semantically meaningful, acoustic unit a discourse is made of. Hence, any attempt of inferring evidence from spoken documents by repeating items, has necessarily to focus on the retrieval of word-like entities.

As we approach word discovery as a specific application of seeded discovery, general remarks hold true concerning the differences in goals and methodology with traditional approaches in speech recognition. In ASR, a lexicon of word models is used jointly with language models for transcribing speech in a statistical framework, relying heavily on labeled training data and supervised learning methods. In word discovery, the objective, as opposite to speech recognition, is to build a similar inventory of word units, the repeating ones, as the final output of the system, without any a priori knowledge, modelling or training. The goal is not that of real-time recognition of speech, but rather of extracting salient segments in the form of repeating words.

Very much related to the specific task, is also the time interval a repetitive pattern can be reasonably expected to repeat. In real spoken contents, grammatical entities like articles, or prepositions occur very frequently, even within the same phrase; but also more semantically significant patterns, like terms linked to a specific topic discussed, are expected to *locally* repeat, where locally might be reasonably quantified in the order of minutes. This is notably different from what can be expected when grossly estimating the average frequency of occurrence of a song in the broadcast schedule of a radio channel.

Specific to word discovery is also the need for appropriate strategies to properly handle the intrinsic variability of speech signal. This high variability makes the task more challenging with respect to other retrieval tasks, like near duplicate discovery. In the following, we enumerate the main sources of variability that are to be considered.

6.1.2 Sources of speech variability

Speech variability represents the main obstacle towards building a robust word discovery system: the algorithm is required to automatically recognize the same linguistic identity in segments possibly being quite dissimilar at the signal level.

Main types of speech variability can be roughly characterized in three categories:

- Intra-speaker variability: the acoustic waveforms produced by the same speaker uttering twice the same word are not identical. This is mainly due to speaking

styles, degrees of co-articulation, health and emotional states of the speaker. These are all factors that might cause acoustic variations in the signal generated. In addition, the type of context might influence the pronunciation: think of different articulation between conversational and reading speech, for instance.

- Inter-speaker variability: simply put, different speakers utter differently. Of course, factors influencing intra-speaker variability might also explains acoustic variations in speech produced by different speakers. In addition, physiological differences between speakers, like vocal tract length are causes of speech variations. Speech uttered by males or females, childs or adults, presents great variation. Notable differences are also noted between native speech and speech coming from foreigners, as well as local accents and speaking styles.

- Environmental conditions: the signal undergoing different channels or subject to different environmental conditions is transformed differently. Ambient noise, characteristics of the room where the speech is produced, like wall thickness and materials, are among the sources of such variability. Furthermore, equipments with which the sound is recorded, and channels where it is transmitted, are also to be kept into account while dealing with speech variability.

In ASR systems different techniques have been proposed to improve the robustness to speech variations. These include:

- the use of larger training databases for better acoustic modelling (see Lamel & Gauvain (2005)).

- front-end techniques for feature normalization: cepstral mean subtraction (Furui (2008)), RASTA filtering (Hermansky & Morgan (1994)) or vocal tract length normalization (VTLN) (Welling *et al.* (2002)).

- adaptation techniques for acoustic models (see Zavaliagkos *et al.* (1995)), but also language models (Seneff & Wang (2005), Huet (2007); Lecorvé *et al.* (2008)).

It should be evident by now that achieving a comparable level of tolerance to variability is even more difficult in unsupervised word discovery, as ASR methods are not applicable, or not straightforwardly applicable, given the assumptions and objectives of our task.

In the following, we start by illustrating the experimental protocol and respective results in a word discovery task by seeded discovery. Later on, extensions on this

baseline system will be presented, specifically intended to cope with variability issues to further improve robustness.

6.2 Experimental set up

The experimental set up is illustrated by detailing three main aspects: a) the data set used for the experiments, the values set for the main parameters, and the performance indicators to rigorously assess the behavior of the system.

6.2.1 Data and main parameters

Test data. Throughout the thesis, the data used for word discovery experiments comprises a subset of the corpus developed for the *ESTER* evaluation campaign for the rich transcription of French broadcast news (Galliano *et al.* (2005)). Each file has been recorded in wave format 16kHz 16-bits from a standard audio card on a PC without any compression. The advantage in using such data is that useful annotations are available, in the form of automatically derived phonetic alignments and various information on speakers: speaker identity (when available), their gender, whether or not they are native French speakers, speaker turns and durations. The availability of phonetic alignments permits to associate an audio segment with the respective string of phonemes; at the evaluation level, this allows to check whether two acoustic repetitions are effectively similar by comparing the corresponding strings, where quantifying similarities is indeed much easier and less ambiguous. In Section 6.2.2, it will be explicitly reported on the use of these phonetic alignments to extract precision and recall measurements for the repetitions found.

For the transcribed part of this corpus, speech is said to account for 97% of the signal, music for 2.3%, the rest being pauses. Thus, this data set is particularly suited for a discovery task focused on the retrieval of words. Moreover, the presence of different speakers, and respective annotations are helpful in assessing speaker dependency. In addition, data includes, at times, telephonic conversations as well as speech mixed with background sounds like jingles, that might enable at least some qualitative remarks on the sensitivity to this type of variability.

In this section word discovery experiments are performed on a 2h speech recording. The first hour is a recording of a radio news show from the French radio channel *France Inter*, broadcasted on April, 18, 2003 from 7 to 8 p.m. The second one is recorded from the same channel, from 8 to 9 p.m.. The size of the test data is competitive

with that of related work in word discovery: in Park (2006), the largest test data is a 1h and a half long recording of an academic lecture. Coming from the same day, news reports in the two recordings mostly refer to similar topics, thus presenting several repetitions of topic-specific terms; this is an attractive property for evaluating the capability of the algorithm in discovering and collecting motif occurrences at the word level.

Main parameters. Main parameters to set in seeded discovery are the length of the seed and respective search buffer, and the value of the spectral threshold ϵ (occasionally called ϵ_{DTW} hereafter) for similarity detection. The length of the seed is set to 0.25 seconds. Accordingly, the motif minimum length is set to 0.5 seconds: we have empirically found that this value represents a good trade off between two distinct requirements: a good word coverage, that pushes toward using a seed as small as possible, and the need to avoid trivial matches, like repetitions at the subword level, that one may encounter while searching for shorter patterns.

The length of the search buffer has been set to 90 seconds; this is assumed to be a reasonable length for the average duration of a news report, where most of topic-related terms are expected to occur. As far as the threshold is concerned, several miss and trial experiments were conducted to tune this parameter by directly comparing different occurrences of the same word, and different words. Nonetheless, experiments are performed while varying the threshold in a range of reasonable values, to study the sensitivity of the algorithm to this critical parameter.

Before describing the actual experiments, performance measures are described, including definitions of precision and recall.

6.2.2 Performance measure.

Assessing quantitatively the performance of the algorithm permits to provide a measure of goodness of the algorithm and to compare objectively different systems, or the impact of the various parameters and features within the same architecture.

Evaluating a motif discovery system amounts to evaluate its output, that is the set of all the acoustic occurrences collected for each motif discovered. One possible way to carry out the evaluation consists in directly listening to all acoustic excerpts and verify that they are effectively occurrences of a same motif. However, human

evaluation is tedious, error prone, and time consuming. A tool is then needed for automatically deriving the required measures.

Phonetic alignments. This goal can be achieved by relying on the annotations included in the ESTER corpus, in the form of word transcriptions and phonetic alignments. This permits to associate the acoustic patterns found with the corresponding words and phonetic strings. While evaluation at the word level is desirable, it is also difficult to perform in practice; this stems from the fact that the algorithm, dealing with an unsegmented stream, rather than extracting word patterns defined by their exact endpoints, may detect short multi-word phrases, subword patterns, or words preceded and followed by neighboring phones, but also *non linguistic* patterns (silences, breathings). In alternative to word level evaluation, performance can be carried out at the phonetic level, by comparing the respective phonetic strings. This can be accomplished by using popular string matching techniques and edit distance measures, where quantifying similarity, for a machine, is definitely easier and less ambiguous than in the acoustic domain.

We remark that the use of phonetic information is solely instrumental to an automatic evaluation of the system, which, in practice, is required to operate uniquely on the audio signal for discovering motifs.

Precision-recall. In many retrieval and recognition tasks, performance are often measured in terms of precision and recall: translated into our specific context, precision (or purity) has to quantify the capability of the system of detecting true instances of a motif and discarding false ones; recall quantifies the capability of collecting, for each motif, as many true hits as possible. In word discovery, such measures are not defined in a standard way; in alternative, by relying on the phonetic alignments in the ESTER corpus, apposite precision and recall measures have been defined at the phonetic level.

Once the passage at the phonetic domain is justified, precision and recall are to be defined from the set of phonetic strings associated to the acoustic occurrences of a motif. Suppose to be able to score the dissimilarity of a pair of such sequences by some string matching technique. Different strategies are then possible for fusing the pairwise scores and defining a unique measure of purity.

Among these possibilities, we opted to mimic the median modelling solution used by the algorithm at the acoustic level, by identifying a median occurrence (or motif

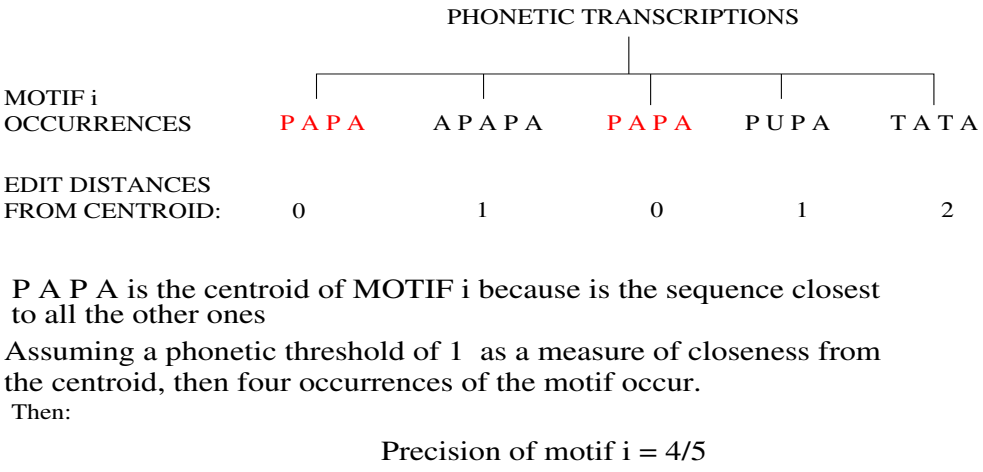

Figure 6.1: Example picturing the selection of the phonetic median occurrence and the computation of precision, according to the edit distance.

centroid) at the phonetic level. This string is defined as the closest, in average, to all the other strings, according to some phonetic distance. The precision for a given motif is then the fraction of the phonetic sequences sufficiently close to the centroid (that is, whose distance from the centroid lies within a specified value, see example in Fig. 6.1). The set of these sequences is a subset of the set of all strings in the corpus that are close to the centroid: the ratio of their cardinality defines the recall for the given motif.

Precision-recall: formal definition. We define these quantities more formally by introducing the following notation:

- LB_i: i-th motif of the library LB.
- $LB_{i,j}$: phonetic transcription of j-th occurrence for the motif LB_i.
- m_i: cardinality of LB_i (the number of occurrences detected for the i-th motif).
- $d(LB_{i,j}, LB_{i,k})$: distance between $LB_{i,j}$ and $LB_{i,k}$. The distance adopted for scoring similarity of strings is the *normalized edit distance* as defined in (Marzal & Vidal (1993)) and implemented in (Vidal *et al.* (1995)), that guarantees a score independent of the length of the strings compared.
- c_i: centroid of LB_i

The centroid c_i of LB_i is defined as:

$$c_i = LB_{i,p} \text{ where } p = \arg\min_{1 \leq j \leq m_i} \sum_{k=1}^{m_i} d(LB_{i,j}, LB_{i,k}) \tag{6.1}$$

The precision of the i-th motif is thus computed as:

$$P_i(\theta) = \frac{\left(\sum_j \delta\left(d\left(LB_{i,j}, LB_{i,p}\right) < \theta\right)\right)}{m_i} = \frac{m'_i}{m_i} \tag{6.2}$$

where $\delta = 1$ if its argument is true, and 0 otherwise. It represents the fraction of instances $LB_{i,j}$ included in a sphere of center c_i and radius θ.

Let m''_i be the number of strings M over the entire phonetic alignment corpus such as $d\left(M, LB_{i,j}\right) < \theta$. The recall of the i-th cluster is the ratio:

$$R_i(\theta) = \frac{m'_i}{m''_i} \tag{6.3}$$

The global precision $P(\theta)$ and recall $R(\theta)$ are computed by averaging P_i and R_i over all motifs in the library. In practice, though, recall is computed only over the sufficiently pure motifs (bearing a precision greater than or equal to 0.5), while the others are simply discarded. The idea is that computing a recall for a motif makes sense if the identity of a motif can be clearly determined from the set of occurrences collected (hence, the condition on the minimum precision required).

Upper bound of the true recall It is worth noting that the definition of recall in Eq. (6.3) accounts only for the motifs discovered by the algorithm at the acoustic level; this measure does not consider those repeating patterns that have not been discovered at all, that should each contribute with a recall $R_i = 0$. Not only, because of convenience in the implementation, we have actually computed the term m''_i by collecting the occurrences of those $LB_{i,j}$ such that $d\left(LB_{i,j}, c_i\right) < \theta$. This set represents a subset of the strings obeying to $d\left(M, LB_{i,j}\right) < \theta$, which makes the respective R_i an upper bound of the true one. The recall measures that will be provided are then too optimistic in this regard.

6.3 Results and discussion

Several experiments were conducted on the 2h test data for different setting conditions. The algorithm has been tested for five different values of the spectral threshold

$\epsilon_{\text{DTW}} = [1.2, 1.4, 1.6, 1.8, 2.0]$ This range of values has been tuned by comparing several occurrences of the same word, for several different words; in principle, tuning or adjusting parameters prior to the discovery violates the unsupervised paradigm of the framework. It might be argued that the algorithm should be capable of automatically tuning and refining on fly these parameters.

In addition to different threshold values, the average and median modelling were each tested to evaluate representativeness and robustness of motif models in different settings.

6.3.1 Quantitative results and impact of modelling

Besides precision and recall, the evaluation is carried out also by providing a number of complementary information, summarized in a series of tables and figures.

Precision and recall measures are reported in Fig. 6.3 where the curves referring to median and average modelling are superimposed on the same plot, for a better visualization. The explicit numbers can be found in Tables 6.1 and 6.2 respectively for the median and average modelling case. In this same tables, additional information is shown by the following statistics (the corresponding abbreviations in Tables 6.1 and 6.2 are also reported):

- Number of motifs discovered (Nm).
- Number of motifs discovered for which a phonetic transcription is available (Nmt).
- Number of motifs yielding a precision greater than or equal than 0.5 (Nocc).
- Average number of occurrences per motif (Nocc/m).
- Number of speakers per motif (Nsp/m).
- Computation time required to complete the task (CPU).

Number of motifs found. The number of different motifs discovered, obviously increases for progressively larger thresholds: for the median case this number goes from 278 up to 5,177; similarly for the average, it goes from 296 up to 5,847. These numbers are also plotted in Fig. 6.2, where it can clearly be observed that the increased number of discovered motifs is not linearly proportional with the increase of the threshold value.

In the third column of both tables, it is specified the number of motifs for which a phonetic transcription can be retrieved from the corpus. The presence of occurrences not retrieved in the phonetic transcriptions, indicates that non-linguistic motifs have also been found: these are mostly short excerpts of jingles, station signatures, news show theme music that are played in the beginning when a news summary is read. It does not come as a surprise that the percentage of this class of motifs is more significant at low values of threshold (in particular, it goes from 74% up to 93% of motifs found for the median modelling and from 69% up to 92% for the average one). These acoustic patterns have a limited variability with respect to speech patterns, and increasing the threshold has the obvious effect of allowing the detection of the more variable repetitions.

Precision and recall Precision and recall measures are reported in the form of red and green curves (respectively for the average and median modelling) in the plot of Fig. 6.3, and the precise values are also readable in the fourth and fifth columns of the two tables.

An immediate glance at the behavior of the curves confirms the expectation that recall and precision respectively increase and decrease while augmenting the similarity threshold. A close look at the single values shows that, while the performance of the two models do not significantly differ, median modelling performs slightly better. As far as precision is concerned, it can be observed that:

- average modelling reports slightly better results at low values of threshold with respect to median modelling: at $\epsilon_{\text{DTW}} = 1.2$ and 1.4 these values are equal to 0.65 and 0.51 for average modelling, they are equal to 0.63 and 0.5 for median modelling.

- At the highest values of ϵ_{DTW}, this behavior is inverted: at $\epsilon_{\text{DTW}} = 1.8$ and 2.0, precision is 0.26 and 0.16 for average modelling, while is slightly better for the median modelling, respectively 0.27 and 0.17.

While these numbers are still similar, it does not come as a surprise that average modelling is more sensitive to a variation of ϵ_{DTW} than median one. At the highest values of ϵ_{DTW} detection of false hits is more likely; whenever occurring, the model resulting from vector averaging is corrupted, in the sense that its representativeness of the initial motif is diminished. This may produce subsequent errors in similarity

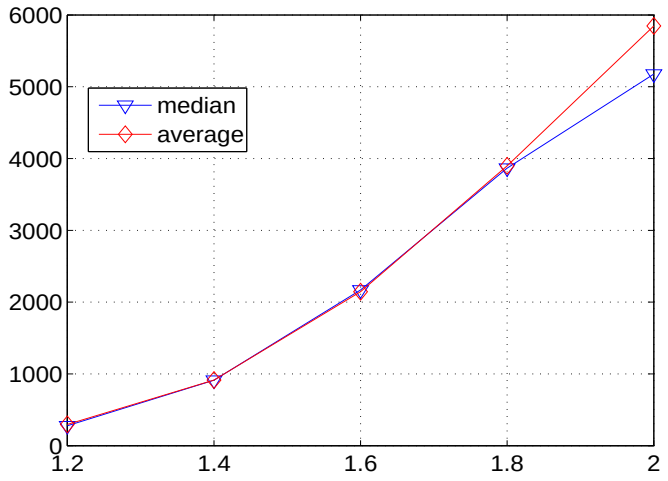

Figure 6.2: Number of motifs found by seeded discovery on the 2h *France Inter* speech file, for average and median modelling and five different values of spectral threshold.

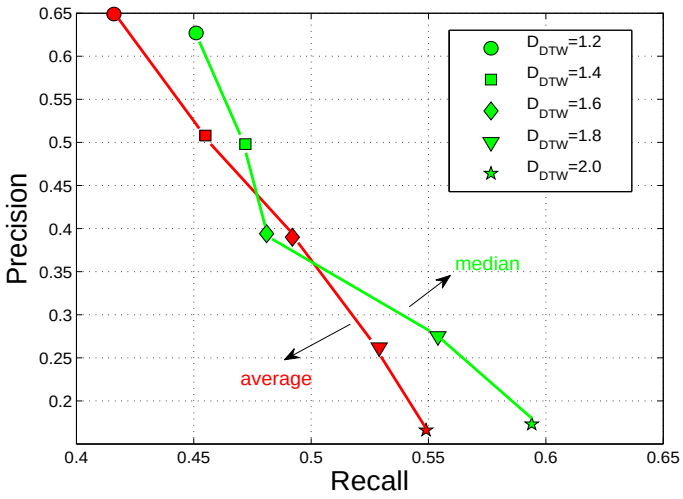

Figure 6.3: Precision and recall measures for seeded discovery on the 2h *France Inter* speech file. The red and green curve represents respectively the average and median modelling case, for five different values of spectral threshold.

detection in the library search stage, leading to poor results in both precision and recall.

Table 6.1: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 2h speech stream. Median modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.2	278	207	0.627	0.451	117	2.051	1.051	44mins
ϵ_{DTW}=1.4	913	777	0.498	0.472	363	2.057	1.036	1h23mins
ϵ_{DTW}=1.6	2169	1948	0.394	0.481	728	2.125	1.115	2h38mins
ϵ_{DTW}=1.8	4187	3866	0.275	0.554	952	2.201	1.259	3h38mins
ϵ_{DTW}=2.0	5177	4808	0.173	0.594	857	2.222	1.6	4h18mins

Table 6.2: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 2h speech stream. Average modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.2	296	204	0.649	0.416	120	2.05	1.033	43mins
ϵ_{DTW}=1.4	915	747	0.508	0.455	359	2.11	1.06	1h24mins
ϵ_{DTW}=1.6	2146	1882	0.390	0.492	723	2.123	1.15	2h42mins
ϵ_{DTW}=1.8	3897	3540	0.262	0.529	981	2.15	1.27	4h18mins
ϵ_{DTW}=2.0	5847	5391	0.166	0.549	1061	2.14	1.48	5h49mins

Median modelling, instead, is less prone to model degradation: if a false instance is collected but the motif median occurrence is still a correct one, then false detection has no impact on model quality. This phenomenon might also be brought up to explain other statistics, as will be shown for recall, number of motifs and number of occurrences and speakers per motif.

Besides precision and recall, a measure of goodness is provided by the percentage of sufficiently precise motifs with respect to the total number of motifs found.

It quantifies the retainable portion of motifs detected, those for which a clear understanding of the underlying linguistic identity can be gathered, the useful output that can be rightfully used in further applications. From the upper to lower end of the threshold range, this percentage declines from 57% to 18% for median modelling, from 58% to 20% for average modelling. In fairness, these values should be higher if one keeps into account the non linguistic patterns that are untranscribed, and not evaluated. For these patterns, precision rate looks extremely high by a gross estimate based on a direct listening; this is likely due to the limited variability of these acoustic segments, that are, for the most, very short pieces of music and jingles, that are easy to recognize as similar by a DTW-based algorithm. This suggests that the algorithm can produce a valuable output for occurrences with limited variability, as increasing the threshold unacceptably damages the purity of the results.

Very much alike, recall rates are affected by the value of the spectral threshold, and thus by the degree of intra-motif variability admitted. As can be observed, in the recall column of tables 6.1 and 6.2, median modelling shows better performance:

- for median modelling, recall goes from 0.45 to 0.59 from $\epsilon_{\text{DTW}} = 1.2$ to $\epsilon_{\text{DTW}} = 2.0$.
- for average modelling, recall goes from 0.41 to 0.55 from the upper to the lower end of the threshold range.

That means that, in average, for high values of similarity threshold, each motif collects the majority of its occurrences in the data (even though, as mentioned, our implementation of recall upper bounds the true one). This improvement, however, is highly detrimental to precision.

The superior performance shown by median modelling is likely to relate to the model degradation issue that impacts also precision values. This same argument can be advocated when explaining the difference in the number of motifs discovered and in the average number of occurrences per motif: model degradation may spread motif occurrences over different motifs, resulting in more motifs, each yielding less occurrences.

Motif length. Not surprisingly, motifs tend to be at the lower end of the length spectrum, in particular in the range $[0.5 - 0.6]$ seconds, where about 80% of the occurrences found fall. This can be concluded by observing Fig. 6.4, where the

distribution of motifs found according to their length is represented. This is expected as in speech data, most repetitions come from words or short multi-word phrases.

Number of occurrences and speakers per motif In the seventh and eight columns of Tables 6.1 and 6.2 the average number of occurrences and speakers per motif is also reported. The numbers, understandably, tend to increase for increasing values of ϵ_{DTW}, and are quite similar for both modelling strategies (the slight difference being likely generated by model degradation issues, as already mentioned).

The intra-motif variability allowed by this range of ϵ_{DTW} is quite limited, as shown by the average number of speakers per motif, always closer to 1 than 2, except at $\epsilon_{\text{DTW}} = 2.0$ for median modelling. As a consequence, the motif discovery system results mostly speaker dependent.

This limited variability influences, in turn, the number of occurrences collected for each motif, the average value being slightly greater than 2 (which is, of course, the minimum number of times a pattern has to occur to be a motif). In fact, the more variable occurrences of a motif (like the inter-speaker ones) tend to be distributed over different motifs.

Further increasing ϵ_{DTW} to account for more variability is not a proper solution, as precision drops steadily making the task unsuccessful.

Computation time Last, the computation time required to complete the task is reported for all the runs of the algorithm, in the last columns of Tables 6.1 and 6.2. There are various factors that impact the computation time, summarized in the following list:

- the length of the search buffer,
- the size of the library,
- the number of candidate matching paths to evaluate by trace back and path extension.

The length of the search buffer is fixed for all the experiments, while the size of the library and the number of candidate motifs are strongly influenced by the value of spectral threshold and modelling strategy adopted. What can be observed from the statistics is that:

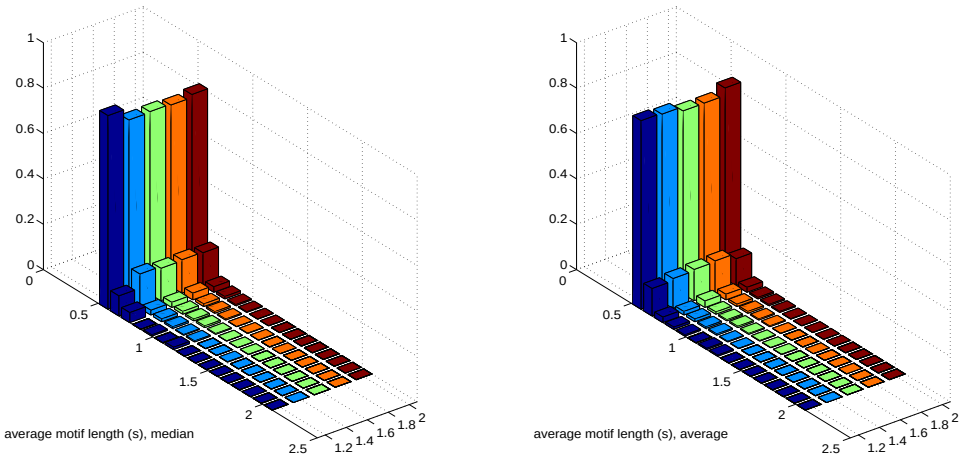

Figure 6.4: Distribution of motif occurrences according to their length for five different values of spectral threshold. Left bar plot: median case. Right bar plot: average case.

1. the computation time grows when increasing the threshold. This is an obvious outcome of the increased number of motifs that are discovered, that implies a more demanding library search.

   For median modelling, the 2h test data is processed in 44 minutes at $\epsilon_{\mathrm{DTW}} = 1.2$, and goes up to about 4 hours at $\epsilon_{\mathrm{DTW}} = 2.0$. For average modelling, computation time ranges from 43 minutes to almost 6 hours.

2. Average modelling requires more time to complete the task with respect to median modelling. This is due to the different library size, as noted already, especially at high values of threshold. In fact, at $\epsilon_{\mathrm{DTW}} = 1.2, 1.4, 1.6$, where the library size is very similar, the computation time does not differ notably.

6.3.2 A qualitative comparison with Park's system

In Fig. 6.5 an example is featured that helps understanding one of the main issues arising from our framework. In this picture an undirected graph is illustrated whose nodes are words and the connecting edges represent the similarity among them.

A false detection occurs as an instance of *France* is erroneously linked to the word *defense*. The other connections are correct and indicate the common belonging to the

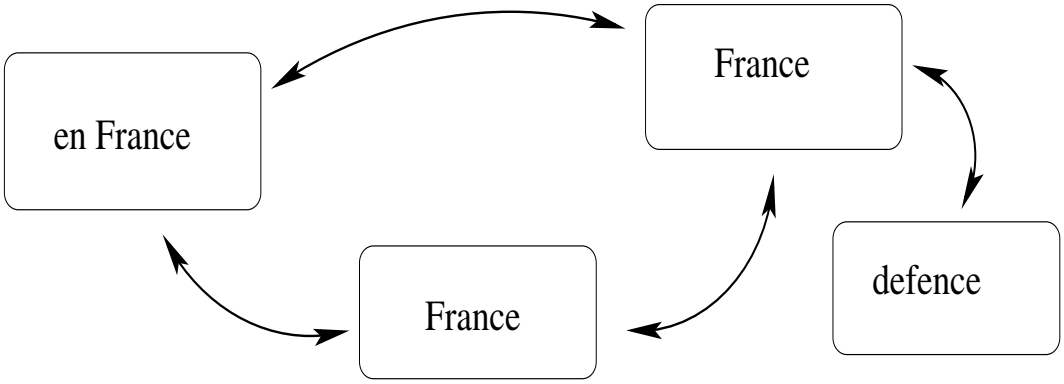

Figure 6.5: Graph representing connections between words deemed as similar.

same motif for the three *France* occurrences (one of them preceded by the preposition *en*). Imagine a situation where the France occurrence linked to *defense* is taken as a seed and matched with *defense* rather than with a nearby *French* occurrence. A motif is initialized and modeled by the corresponding average sequence of vector (if, for instance, average modelling is used). Resulting by the average of two different words, the model is not sufficiently representative of the word *France* to match with the two subsequent occurrences, which, in turn, are too distant in time to even be compared.

In such unfortunate situation, the final outcome is a motif comprising two different words, and a potential motif of three occurrences definitely lost. In this case, failure is to be credited to:

1. The insufficient size of the search buffer.
2. The corruption of a model generated by a false hit.
3. The absence of a recovery mechanism to detect and correct the error.
4. The pattern matching technique (for recognizing as similar two different words).

It is interesting to think of how the framework designed in (Park (2006)) word discovery system might operate in the same scenario. In that system, all speech segments are subject to a pairwise comparison, therefore all the connections are retrieved as in Fig 6.5 (as in our case, similarity score is provided by a DTW distance). In a subsequent stage, clusters are formed by merging densely connected nodes: that might

well permit to group together the three *France* occurrences, each of them connected to the others, and cut the edge between the one *France* occurrence and *defense*, thus allowing for a perfect motif discovery. In this framework, the problem of insufficient search buffer size does not hold, because every speech segment is preliminarily compared with all the other ones; next, model representativeness is not an issue, as no model is created, but all fragments found and the reciprocal similarity are used for clustering; the clustering itself, beside grouping similar occurrences (thus benefitting recall), acts also a secondary pruning mechanism, benefitting precision.

This property makes Park's framework very attractive, but also highlights the different spirit lying at the core of these two contrasting systems. In one case, the batch framework takes advantage of the availability of the whole data for performing pairwise comparison and the subsequent clustering. In the other, sequential processing is assumed as a mandatory requirement for a streaming algorithm that aims at being easily applicable on different tasks, like near duplicate discovery in large streams of composite audio (as it will be illustrated in the next chapter). Besides, Park's computation system implies a quadratic complexity with respect to the number of fragments the data has been segmented into. In fact, it could be considered as an instance of those naive methods depicted by Figure 3.2.

The evaluation of the experiments will be completed in the next section by describing the output of the system from a more qualitative point of view.

6.3.3 Qualitative remarks on the motifs found

The qualitative analysis of the motifs found serves the purpose of providing the reader a more concrete indication of the type of repetitions found and the most common errors encountered.

To this end, we attempt to broadly classify these motifs in:

1. non-speech patterns.

2. speech patterns.

Moreover, while we do not claim that our method selects repetitions according to semantic relevance, we also provide a distinction in semantically meaningful motifs, as to better highlight its potential usefulness in summarization tasks. In fact, the following observations are made by simply scanning the library of motifs and listening to the excerpts found, without any prior knowledge on the content of the file.

Non speech patterns. One class of motifs found, as already mentioned, is represented by non linguistic patterns exhibiting a limited variability. These are short pieces of music, jingles or part of advertisements, whose duration is comparable to that of single words, or short multi-word phrases. As mentioned, these sounds are not transcribed and thus are not accounted for evaluation purposes.

Speech patterns: non linguistic motifs. One type of extremely frequent pattern is represented by the short breathing in between consecutive words. This type of sound is recognized and transcribed by a speech recognizer, thus part of the formal evaluation process. These patterns are noted to be less variable among different speakers: inter-speaker occurrences happen to be detected also at low spectral thresholds.

Speech patterns: semantically relevant words. The algorithm shows the capability of discovering and collecting occurrences of words. Because of the minimum length requirement, these patterns are mostly words comprising several syllables; thus, grammatical entities, like prepositions or articles, which are usually very repetitive but extremely short, do not appear but associated with a following word.

Among those words that help identifying the content of a spoken document, we report the name of the characters involved in a certain news: so the fragments corresponding to French politicians *Nicolas Sarkozy*, *Jean Marie Le Pen*, *Francois Hollande* as well as their location in the data, are indicative that political chronicle is the main subject in that part of the audio recording. The impression is further confirmed when listening to occurrences of *la gauche*, *le partie socialiste*, *front national* (respectively meaning *the left*, *the socialist party*, *national front*, the latter being the name of the party led by Jean-Marie Le Pen). If among those repetitions, also temporal cues are present, like occurrences of the date *vingt et un avril deux mille deux*, French listeners might get a reliable feeling that the news is specifically addressing the presidential elections, held in France on April 21, 2002.

Repetitions of the word group *crise en Irak*, *Amerique* or *americain* (respectively *crisis in Iraq*, *America*, *American*) as well as the name *Saddam Hussein al-Tikriti* might also hint to some news concerning the american military intervention in Iraq against Saddam Hussein.

Speech patterns: semantically irrelevant words Of course, also repetitive patterns that do not bear semantic significance are found, like, for example, *particulierment*, *cet apres-midi*, *la temperature*, *super* (respectively meaning *particularly*, *this afternoon*, *the temperature* and *super*).

As far as the most typical errors encountered, we can grossly attempt to classify them in three main categories:

1. **motifs sharing a common subword**: this kind of error occurs whenever different words that share a common subunit are recognized as similar. As characteristic example, we can mention the several occurrences of words ending in *ation*, like *application*, *mobilisation*, *nation*, *segregation* etc. It is likely that this type of error originates when the matching subsegments are retrieved in the seed match procedure and the partial occurrences are extended from the respective endpoints, adding non matching frames of signal as a prefix and suffix. This may possibly lead to finally collect two sequences that are sufficiently long to meet the minimum length requirement, while only partially matching.

   In the future, we may be able of mitigating these errors by evaluating the distortion profile of the extended path to note the discontinuities generated by the false matches at the boundaries.

2. **acoustically similar but lexically different motifs**: some acoustic sequences are composed by different phonemes, hence bearing a different lexical identity, but their acoustic realization might be quite similar. This strongly depends on pronunciation, and thus speaking style and language. This might be the case of the similarity between the words *poisson* and *pour cent* (effectively found in one experiment), that pronounced by French speakers might present a significant degree of resemblance. The speech recognizer that has produced the phonetic alignment correctly reconstructs the different phonetic composition of the utterances, which reveals how the performance of the our similarity detection technique are not as good as those of a state of the art recognizer, as expected.

3. **completely unrelated sequences**: this happens when different sequences, both at the acoustic and phonetic level, are recognized as occurrences of a same motif. Understandably, this is an error frequently encountered at the

highest values of spectral threshold, and the main reason that prevented a further increase of the similarity threshold in the described experiments.

6.4 Comparison of self similarity matrices and application to word discovery

Experiments in the previous section highlighted that an acceptable level of purity of the motifs found can be reasonably guaranteed by admitting a limited amount of intra-motif variability. This, in turn, impacts negatively recall rates and, in general, the capability of the system to recognize the more variable occurrences of a motif. Increasing the DTW threshold exposes the system to unacceptable false detection rates that determine the failure of the task.

In this section a template matching technique is introduced that aims at improving robustness to speech variability. The core idea is to determine whether two speech sequences are occurrences of a same motif according to the similarity of their self similarity matrices (SSMs). The fundamental assumption is that these matrices carry meaningful information on the acoustic-phonetic structure of the underlying sequences; such information is exploited for recognizing whether the compared acoustic segments share the same lexical identity (hence belonging to the same motif).

While such technique could be directly applied for comparing acoustic sequences, it fully delivers a beneficial impact on performance when used in conjunction with DTW in a two-stage cascade system. SSM comparison is mainly used as a secondary pruning mechanism that selects motif occurrences from a set of motif candidates outputted by the DTW-based stage.

The illustration of the technique and its integration in the system comprises several different steps, namely:

- the description of the basic concepts.
- The practical implementation of the techniques and the definition of dissimilarity measures of SSMs.
- A validation of the method in practical scenarios, respectively in word spotting and word discovery experiments.

6.4.1 Basic concepts

The self similarity matrix (SSM) of a sequence χ_a^b is the square symmetric matrix $\Phi(\chi_a^b)$ defined as $\Phi\,(i,j) = d\left(\chi_a^b(i), \chi_a^b(j)\right)$. It follows the it has a zero diagonal, that is $\Phi\,(i,i) = 0$. In agreement with our choice of using Euclidean distance in DTW, d is defined by this same distance also in SSM computation.

Suppose to decompose a frame χ_i of a sequence χ as:

$$\chi_i = S_i + N \tag{6.4}$$

where S_i carries the *linguistic* information on the frame, and N is the component that accounts for all non linguistic factors (speaking styles, channel, background sounds, etc...), that are to be considered as noise for our purpose.

If such an *additive noise* model were valid, then SSMs would only depend on the linguistic identity of the original sequences, as all the noisy components would cancel each other out in computing $\Phi(i,j)$. For instance, in Cepstral Mean Subtraction (CMS), mean removal follows the belief that a constant additive factor (the DC component) is responsible for channel distortion. However, with regard to the other noisy factors, this property does not strictly hold true if speech is represented by sequences of MFCCs. Moreover, even assuming a perfect accuracy of the additive model in (6.4), comparing the CMS normalized sequences would be more straightforward, rather than comparing the respective SSMs.

The advantage of using SSMs is that the distances among mutual parts of a sequence, generate a two-dimensional pattern that is peculiar of the underlying acoustic-phonetic structure. By comparing SSMs we intend to recognize the recurrence of such patterns among speech sequences to reveal their common belonging to the same motif.

Example Outputs. It is possible to observe consistent similarities of SSMs across several different conditions, that is when instances of a same word are uttered by different speakers, or undergo different channels or are imposed on a noisy background.

As a matter of example, two figures are included showing the self similarity of different utterances of the same word. In Fi. 6.6 the SSMs of four occurrences of *Jean-Marie Le Pen* (a French politician frequently cited in the recording) are shown, the top two from two different male speakers, the bottom two from two different female speakers. Those matrices differ in size (because of different speaking rate),

and in intensity values of single entries (or *pixels*, following a usual image processing nomenclature). However, a clear resemblance can be observed by local edges and shape patterns that ultimately depends on the phonetic identity of the underlying word.

In Figure 6.7, four similarity matrices are shown for different renditions of *vingt-et-un-avril* (April 21th, in English, the day where the French presidential elections were held). The top two are from two different male speakers, the second two from the same female speakers, the second one being superimposed on a musical background (a short jingle). The presence of these noisy patterns makes significantly high the spectral distance between the two sequences, as measured by a DTW dissimilarity measure. However, visual resemblance of the respective self similarity matrices looks striking.

SSMs are thus interesting candidates for robust pattern matching and a distance between SSMs is therefore required.

6.4.2 Definition and implementation

In order to define a distance between SSMs, two main issues are to be accounted for:

- different motifs may also have similar SSMs.
- motif occurrences may have a different length, and the respective SSMs a different size.

SSM-DTW cascade The first issue stems from the fact that similarity of SSMs is a necessary but not a sufficient condition for speech sequences to be deemed as similar. Indeed it may be possible to generate similar SSMs from different words. SSM comparison is an indirect way of quantifying how sequences are related; differently from DTW methods, the comparison is not directly performed over the sequences themselves, but on the self-distance patterns they generate.

To cope with this problem, we propose using SSM-based and DTW-based matching techniques in conjunction. Practically, this is accomplished by allowing more variability in the DTW-based comparison, by setting a higher ϵ_{DTW}. In a subsequent stage, motif candidates determined from the previous stage, are subject to an SSM-based validation; the end goal is to properly filter the false hits that the increased amount of variability admitted is likely to have determined.

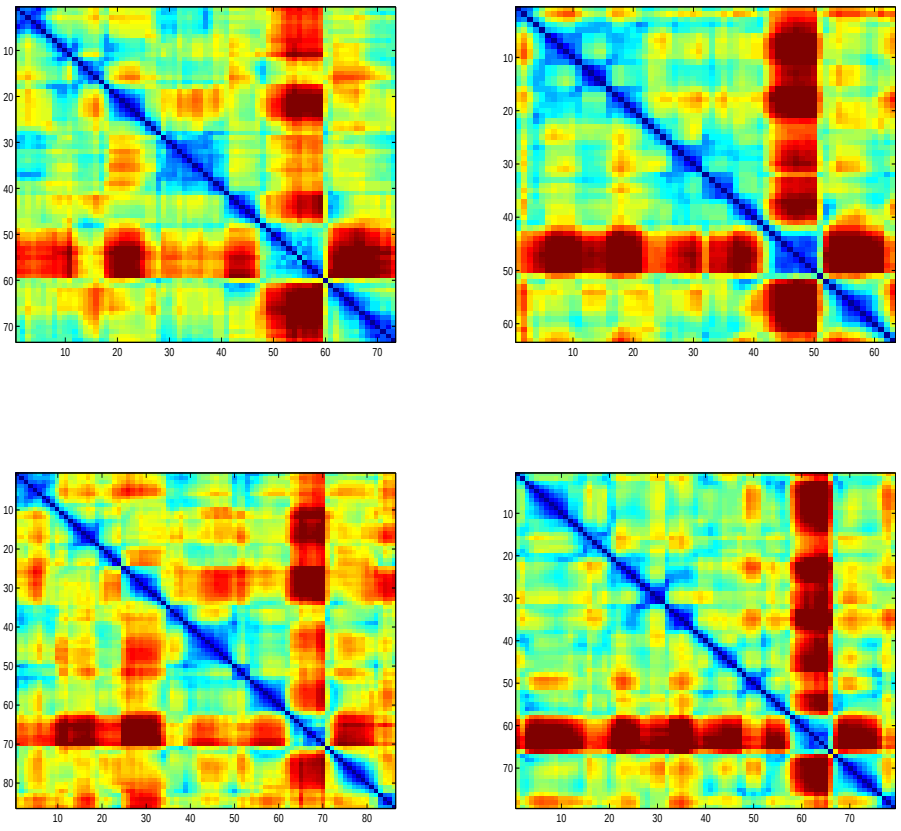

Figure 6.6: SSMs of four occurrences of *Jean Marie Le Pen*. The top two are from different male speakers, the bottom two from different female speakers.

According to this operative framework, two sequences χ_1 and χ_2 are deemed as similar if $D_{\text{DTW}} < \epsilon_{\text{DTW}}$ and $D_{\text{SSM}} < \epsilon_{\text{SSM}}$ (as illustrated in Fig 6.8).

Size normalization To account for the difference in size, the sequences χ_1 and χ_2 are rewarped according to the matching path $P = \{(i_k, j_k)\}_{k=1}^{L(P)}$ found by DTW, to obtain two sequences of the same length $L(P)$.

Formally $\tilde{\chi_1} = \{\chi_1(i_k)\}_{k=1}^{L(P)}$ and $\tilde{\chi_2} = \{\chi_2(j_k)\}_{k=1}^{L(P)}$.

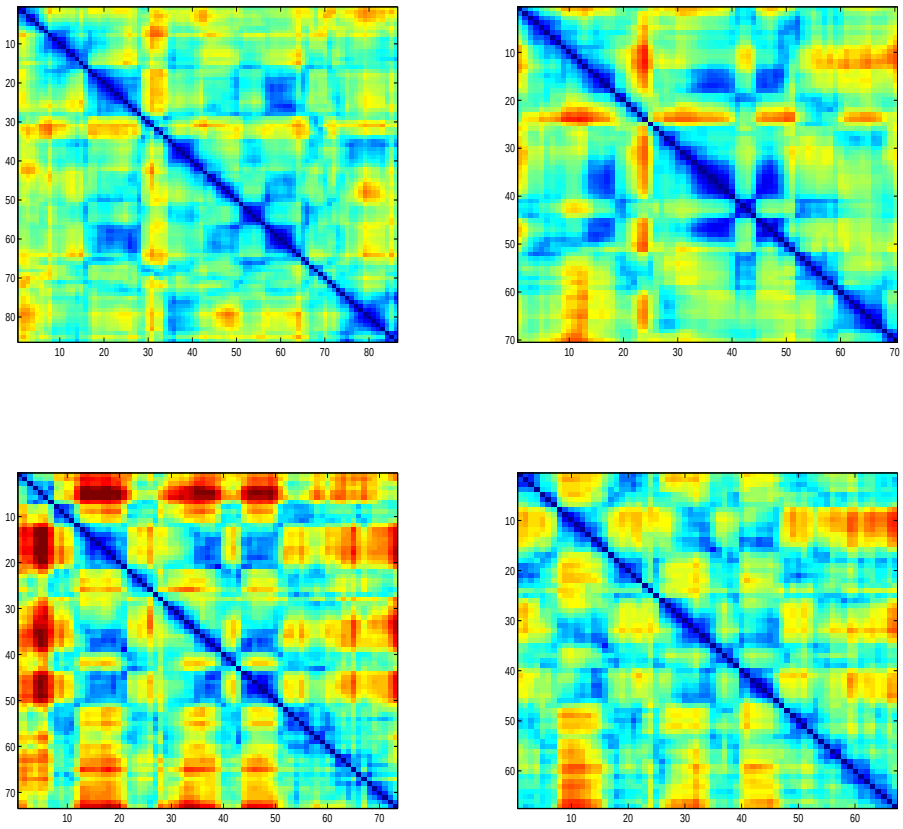

Figure 6.7: SSMs of four occurrences of *Vingt-et-un avril*. The top two are from two different male speakers, the second two from the same female speakers, the second one being superimposed on a musical background (a short jingle).

SSM dissimilarity measures Different metrics are then possible: a simple one is the L_1 norm of the matrix difference normalized by its size, that is $D_{\text{DTW}} = \|\Phi(\tilde{\chi_1}) - \Phi(\tilde{\chi_2})\|_1 / L(P)^2$. This distance strictly depends on the absolute values of the matrix entries and does not encode well the implicit spatial pattern.

Therefore, we propose an additional distance based on the computation of local *histograms of oriented gradients* from the SSMs (see Dalal & Triggs (2005)), effectively

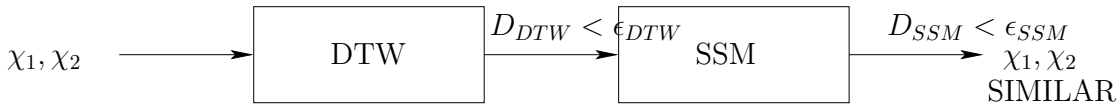

Figure 6.8: Cascade of DTW and SSM comparisons to recognize similarities between segments.

regarded as gray scale images. The underlying assumption is that local objects' appearances and shapes can be well characterized by the distribution of local intensity gradients and edge orientations.

The practical implementation is performed by dividing the matrix into a dense, uniform grid of overlapping square regions (*blocks*), divided in turn into smaller square regions (*cells*). For each cell, we compute a local 1-D histogram of gradient directions over the entries (or *pixels*) of the cell, weighted by the gradient magnitudes. The final descriptor is the concatenation of the normalized histograms over all the cells. The distance between descriptors extracted from SSMs is the normalized norm L_1 of the vector difference, indicated as D'_{SSM}. Following recommendations in Dalal & Triggs (2005), we opt for the following parameters (see also Fig 6.9):

- 3×3 blocks of 12×12 pixels as a local patches for histogram computation.
- 0.5 overlap between block in both horizontal and vertical direction.
- uniform histogram with 18 bins in the $[0^\circ - 360^\circ]$ range.
- $[-1, 0, 1]$ gradient filter with no smoothing for the computation of the 1-st order derivatives in the horizontal and vertical direction.

Note that these two distances provide complementary information on the structure of the SSM.

The first one measures directly the difference in the intensities of the entries, assuming that, across different conditions, occurrences of a same motif should preserve the mutual distances of their parts (this is true if the additive noise model were exact).

The second one relaxes that condition, since descriptors do not depend on pixels' magnitudes but rather on local gradients' magnitudes (i.e. they measure the strength and directions of local edges). Moreover, they do not provide just a punctual information (i.e. confined to each single pixel) but are computed over dense, overlapping

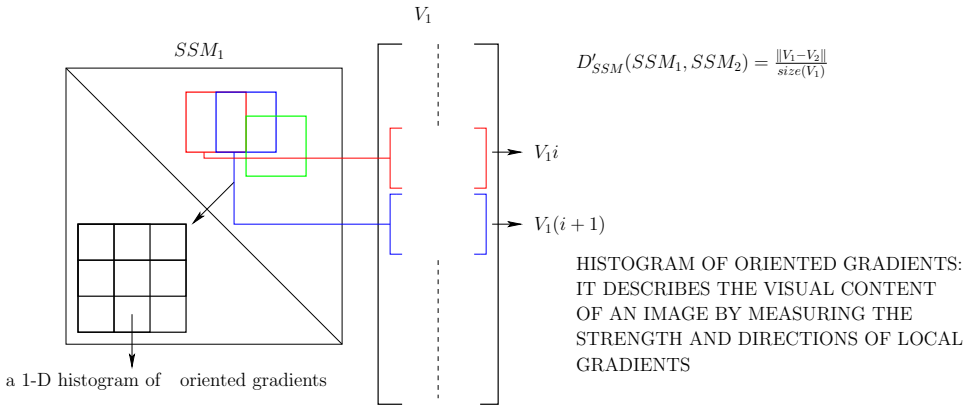

Figure 6.9: Practical implementation of the histogram of oriented gradients techniques

grids of pixels, encoding a richer and more complex information on the self-similarity visual patterns.

It can be then concluded that two sequences χ_1 and χ_2 for which the relation $D_{\text{DTW}} = (\chi_1, \chi_2) < \epsilon_{\text{DTW}}$ stands, are validated as similar if satisfy the two conditions $D_{\text{SSM}}(\chi_1, \chi_2) < \epsilon_{\text{SSM}}$ and $D'_{\text{SSM}}(\chi_1, \chi_2) < \epsilon'_{\text{SSM}}$.

6.4.3 Application to word spotting

We consider as a data set an audio file of about 20 minutes built by concatenating six short excerpts of various broadcast news shows from the ESTER corpus. The news shows all focus on the upset behind the success of French politician Jean Marie Le Pen, at the first round of the presidential elections held on April 21st, 2002. Thus, the presence of different speakers of different gender talking on the same topic, makes this data set particularly suited for experiments aimed at assessing the impact of the SSM comparison stage.

We have chosen four different words (or small multi-words expressions) repeating several times and bearing also significant semantic relevance in relation to the topic discussed: Jean Marie Le Pen, vingt-et-un avril, extreme droite, France.

In particular, we have randomly extracted one occurrence of each keyword and searched for all possible repetitions in the speech recording. The search is conducted by relying on the SLNDTW algorithm to select candidate motifs. Each pair of segments (the keyword template and its hypothesized repetition), undergo subsequently

Table 6.3: Precision-Recall values for keyword spotting experiment with DTW and SSM combined

keyword	D_{DTW}	$+D_{\mathrm{SSM}}$	$+D'_{\mathrm{SSM}}$
Jean Marie Le Pen	P=10/30, R=10/17	P=10/25, R=10/17	P=10/18, R=10/17
Vingt-et-un Avril	P=12/68, R=12/17	P=12/55, R=12/17	P=12/28, R=12/17
Extreme Droite	P=4/23, R =4/7	P=4/16, R=4/7	P=4/6, R=4/7
France	P=10/91, R=10/23	P=9/51, R=9/23,	P=8/36, R=8/23

an SSM-based comparison, based either on the sole ϵ_{SSM} threshold or on both thresholds, ϵ_{SSM} and ϵ'_{SSM}. The goal is to evaluate the effect of adding the SSM validation to precision and recall; it is clear that, by using SSM as a validation stage operating only on the output of the preceding DTW-based comparison, recall rate cannot improve, as no additional true occurrences can be detected. The hope in using SSM is then to improve precision, by recognizing and discarding the false hits collected, while leaving recall unchanged, by rightly retaining the true hits detected by SLNDTW. Results in terms of precision and recall are reported in Table 6.3 when only DTW is used, and when the two SSM techniques are then added. The numbers are to be read in the following manner: each precision-recall value is reported as the ratio of two values. The numerator, for both precision P and recall R, indicates the number of true repetitions of the keyword found with the specified technique. The denominator indicates, for the precision, the total number of occurrences collected, and for the recall, the number of true occurrences of the keyword in the audio data. We have used on purpose a DTW threshold $\epsilon_{\mathrm{DTW}} = 3.0$ very high in order to allow for more variability among motif occurrences. As a matter of fact, the precision is very poor when only DTW is used for similarity detection, while the combined use of the SSM techniques leads to a substantial improvement in the precision rate while keeping the recall untouched for all but the shortest keyword *France* (where also two correct occurrences have been erroneously discarded). Using a high spectral threshold, besides, has allowed to find several inter-speaker occurrences, up to six different speakers, both male and female. We will see similar results when applying the technique to the more challenging word

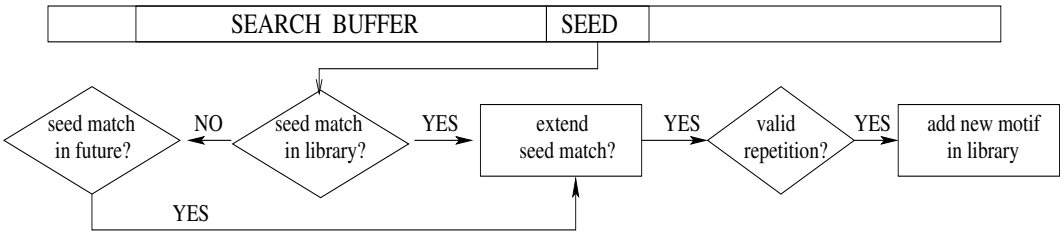

Figure 6.10: Modified architecture of seeded discovery: the validation stage comprises the SSM comparison. If occurrences resulting from seed match extension are further deemed as similar by SSM comparison, a pair of matching segments is detected.

discovery task, on a larger data set.

6.4.4 Application to word discovery

Results on the use of self similarities in a word spotting encourage their application in a more challenging context. In this section the baseline architecture of seeded discovery is modified to include the self similarity comparison as an additional validation stage (a visual illustration is presented in Fig. 6.10).The validation stage can be viewed as the similarity score subtask, as candidate motifs undergo a further comparison after the preliminary selection operated in the DTW stage (which corresponds to the similarity detection subtask).

6.4.4.1 Data and main parameters

Test data. The test data is represented by a 4h speech file. The first 2h are the same of the experiment detailed in paragraph 6.2.1, two *France Inter* news shows recorded on the same day. The last 2h are two additional news shows, each of the duration of 1h, recorded from the channel *France Info* the same exact day of the first two.

Testing the algorithm on such data is particularly challenging because of two main reasons:

- many topic-related terms occur, since news shows aired the same day, thus likely covering similar subjects.
- coming from different hours of the day and two different channels, those terms are uttered by multiple speakers. This is useful for testing the sensitivity to

the speaker dependency issue and the capability of the system of dealing with a high degree of variability.

Main parameters. The seed and search buffer lengths are left unchanged with respect to the configuration of the experiments in Section 6.2.1: the seed length is 0.25 seconds and the search buffer length is 90 seconds.

The algorithm is run for six different values of spectral threshold, precisely $\epsilon_{\text{DTW}} = [1.7, 1.9, 2.1, 2.3, 2.5, 2.7]$. These values are higher than those employed in the previous experiment, as we explicitly intend to check the performance when more spectral variability is admitted.

Moreover, with respect to the previous set up, we have opted for a slightly different weight of the diagonal move. If $k_{h,v}$ indicates the weight for the horizontal and vertical moves, and k_d for the diagonal one, we set $k_{h,v} = 1$ and $k_d = 0.7$, instead of 0.5 as previously done. This weight favours less the diagonal move as it was noted to better model the differences in speaking rates when the same word is uttered by different speakers. This is of utmost importance because the end goal is to be also able of correctly identifying inter-speaker occurrences of a motif.

Concerning the values of ϵ_{SSM} and ϵ'_{SSM}, we have fixed a unique value for all the runs of the algorithm, tuned by apposite miss and trials experiments.

6.4.4.2 Results and discussion

The quantitative results are summarized in a series of figures and tables. In Fig 6.11 and 6.12, precision-recall curves for the SSM provided system are shown together with those obtained by the DTW based system, for direct comparison. The first figure refers to performance obtained by median modelling, the second one refers to the average modelling case.

The single values of each performance indicator are reported in the four tables 6.4, 6.5, 6.6 and 6.7. The first two report statistics for the median modelling case, respectively for the DTW+SSM system and the DTW-based system alone. The last two tables report these same statistics for the average case, for the DTW+SSM system and the DTW-based system alone.

Precision and recall. The curves in Fig. 6.11 and 6.12 clearly show the benefit of the joint use of DTW and SSM-based pattern matching technique.

This can be observed by comparing precision and recall values for the two systems, for a given modelling strategy and degree of spectral variability admitted. For instance, in the median modelling case, precision of the SSM-provided system constantly outperforms of about 10% the DTW based one, at an almost constant recall rate (specific numbers are reported in the fourth and fifth columns, respectively for precision and recall, of Table 6.4 and 6.5). A similar case can be made concerning precision-recall performance when average modelling is employed. Here the improvement in precision amounts to about $10-15\%$, with respect to the DTW based system, at an almost (or even slightly better) recall rate (see specific numbers in respective columns in Table 6.6 and 6.7).

As already explained in the word spotting experiment, such improvement is due to the smart pruning action performed by the SSM-based validation stage. Improving precision while leaving recall basically unchanged, amounts to admitting that SSM is successful in discriminating true and false hits from the set of motif candidates hypothesized by the preceding DTW-based detection stage.

This same trend, and for this same reason, can be noted by observing the portion of sufficiently precise motifs: it is the ratio between the number of motifs yielding a precision greater than or equal to 0.5 (Nocc in the sixth column of all tables) and the number of motifs found for which a phonetic transcription is available (Nmt in the third column of all tables). This percentage goes from 61% for the SSM case against 49% for the DTW one at $\epsilon_{\text{DTW}} = 1.7$, down to 26% against 16% at $\epsilon_{\text{DTW}} = 2.7$. Similarly for the average modelling, it goes from 63% for the SSM provided system against 46.7% for the DTW-based system at $\epsilon_{\text{DTW}} = 1.7$, down to 26.6% and 15.1% at $\epsilon_{\text{DTW}} = 2.7$.

Number of occurrences and speakers per motif We can further comment on the behavior of the two different systems by observing two additional performance indicators, and namely: average number of occurrences and speakers per motifs, reported in the seventh and eight columns of the tables.

For a given modelling, at the same value of ϵ_{DTW}, the value computed for both parameters is always slightly higher for the SSM-supplied system with respect to the DTW-based system. This slight difference is hardly surprising, as reflected also by the negligible difference in recall, that is tightly related with these two performance measures. In fact, the primary merit of the joint use of DTW and SSM is to allow for a superior purity of the results, for a given amount of intra-motif variability

admitted by ϵ_{DTW}. The slightly greater values of recall, and number of speakers and occurrences per motif are likely due to model degradation issues, that are related, in turn, to purity. In fact, as we have already noticed in the previous section, the purity of the result allows to preserve the quality of the model in terms of representativeness of the underlying motif, which is essential for correctly recognizing additional motif occurrences.

Computation time In the last column of Tables 6.4, 6.5, 6.6 and 6.7, the computation time required to accomplish the task for the different systems and modelling strategies is reported.

The numbers are quite similar, for a given modelling strategy and ϵ_{DTW} value, even when moving from the DTW-based system to the SSM-based one. For instance, from Table 6.4 and 6.5, it can be observed the similarity of the computation times at $\epsilon_{\text{DTW}} = 1.7, 2.1, 2.5$ for the two system, in the median modelling case. At $\epsilon_{\text{DTW}} = 2.7$ the SSM-based system is three hours slower than the DTW-based one, while, on the other hand, it performs the computation three hours faster at $\epsilon_{\text{DTW}} = 2.3$.

Similar observations can be made by considering the value of this same performance indicator for the average modelling case. Here computation times looks very similar for all threshold values but $epsdtw = 2.5$ where the SSM based one is almost four hours slower.

The justification of this result is straightforward: while the use of the SSM validation stage adds a further computational burden in the system, the reduced library size (as determined by the pruning action of SSM) results in a less demanding library stage.

This further highlights the attractiveness of this additional feature, as computation time does not look significantly affected.

6.5 Summary

In this chapter, we have then shown the applicability of seeded discovery to a word discovery task. We have preliminarily introduced novel precision-recall measures at the phonetic level to assess results from a quantitative point of view. Afterwards, through a series of experiment, we have shown that DTW-based seeded discovery is capable of extracting repetitions of words and word-like patterns when a limited degree of variability is enforced, resulting in a substantial speaker-dependent system.

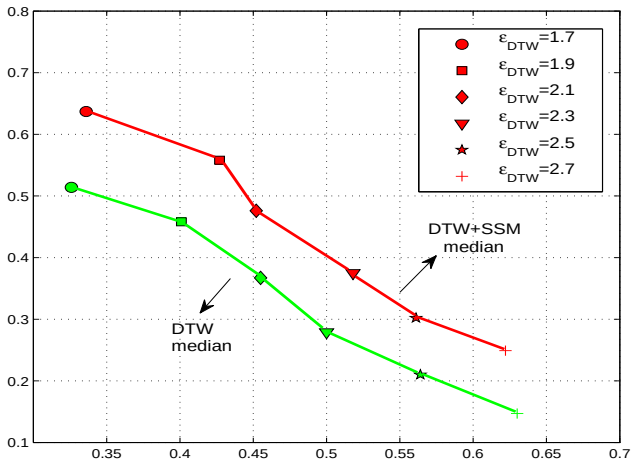

Figure 6.11: Precision and recall measures for seeded discovery on the 4h *France Inter* speech file. The red and green curve represents respectively the DTW+SSM and DTW alone system, for median modelling and six different values of spectral threshold.

To partially mitigate these issues, we have introduced a template matching technique based on the comparison of the self similarity matrices of speech sequences. The idea is to investigate the spatial structure of these matrices, effectively seen as gray scale images, to recognize a two dimensional pattern dependent on the acoustic phonetic identity of the underlying sequences. We have shown, in word spotting and word discovery experiments, that the joint use of DTW and SSM based comparison, is beneficial for improving robustness of the system to speech variability. The benefit consists in the possibility of admitting an increased amount of spectral intra-motif variability, to allow for the detection of the more variable occurrences of a speech motif (like the inter-speaker ones), while ensuring an acceptable level of purity of the repetitions detected, at an almost constant recall rate.

In the following chapter, we extend the applicability of seeded discovery while describing its usage in a near-duplicate discovery task, and more specifically, in song discovery experiments.

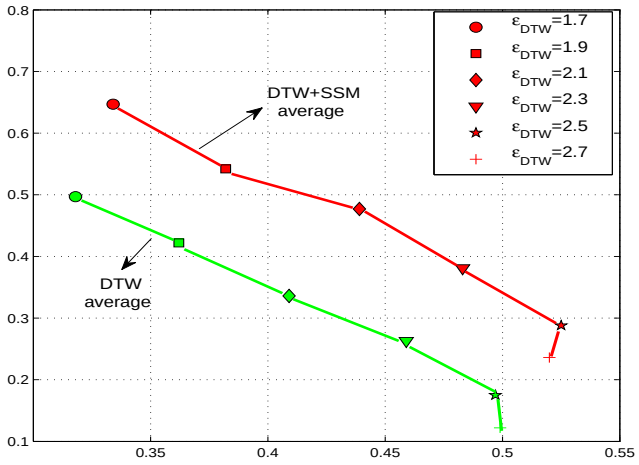

Figure 6.12: Precision and recall measures for seeded discovery on the 4h *France Inter* speech file. The red and green curve represents respectively the DTW+SSM and DTW alone system, for average modelling and six different values of spectral threshold.

Table 6.4: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 4h speech stream. Median modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.7	600	447	0.637	0.336	275	2.127	1.106	2h32mins
ϵ_{DTW}=1.9	1222	1013	0.558	0.427	551	2.290	1.136	6h40mins
ϵ_{DTW}=2.1	2120	1860	0.476	0.452	878	2.309	1.215	8h00mins
ϵ_{DTW}=2.3	3163	2851	0.375	0.518	1072	2.375	1.37	12h24mins
ϵ_{DTW}=2.5	4113	3763	0.302	0.561	1177	2.398	1.52	17h57mins
ϵ_{DTW}=2.7	4896	4517	0.249	0.622	1183	2.43	1.68	23h2mins

Table 6.5: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 4h speech stream. Median modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.7	860	651	0.514	0.326	319	2.138	1.106	2h40mins
ϵ_{DTW}=1.9	1818	1551	0.458	0.401	706	2.198	1.136	5h23mins
ϵ_{DTW}=2.1	3244	2922	0.367	0.455	1049	2.261	1.166	7h59mins
ϵ_{DTW}=2.3	4985	4593	0.279	0.500	1306	2.336	1.251	15h36mins
ϵ_{DTW}=2.5	6488	6037	0.210	0.564	1341	2.310	1.357	18h26mins
ϵ_{DTW}=2.7	7036	6562	0.147	0.63	1383	2.26	1.480	20h18mins

Table 6.6: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 4h speech stream. Average modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.7	655	455	0.647	0.334	287	2.132	1.105	4h55mins
ϵ_{DTW}=1.9	1302	1040	0.542	0.382	562	2.153	1.168	7h38mins
ϵ_{DTW}=2.1	2256	1916	0.477	0.439	909	2.172	1.236	14h13mins
ϵ_{DTW}=2.3	3366	2967	0.381	0.483	1170	2.182	1.312	15h04mins
ϵ_{DTW}=2.5	4519	4058	0.288	0.525	1268	2.187	1.426	22h36mins
ϵ_{DTW}=2.7	5493	4944	0.236	0.520	1314	2.182	1.470	27h50mins

Table 6.7: Number of motifs (Nm), Number of transcribed motifs (Nmt), Number of motifs yielding an acceptable precision (Nocc), average number of occurrences per motifs (Nocc/m), average number of speakers per motifs (Nsp/m) (these last two computed over the Nocc motifs) and CPU time for different values of spectral threshold on the 4h speech stream. Average modelling.

threshold	Nm	Nmt	Prec	Recall	Nocc	Nocc/m	Nsp/m	CPU
ϵ_{DTW}=1.7	1009	700	0.497	0.318	327	2.165	1.084	4h34mins
ϵ_{DTW}=1.9	2100	1666	0.422	0.362	692	2.193	1.102	8h22mins
ϵ_{DTW}=2.1	3632	3119	0.336	0.409	1071	2.210	1.166	14h14mins
ϵ_{DTW}=2.3	5560	5064	0.263	0.459	1428	2.208	1.215	15h38mins
ϵ_{DTW}=2.5	7996	7408	0.175	0.497	1501	2.175	1.305	18h52mins
ϵ_{DTW}=2.7	10457	9724	0.122	0.499	1473	2.143	1.383	27h40mins

Chapter 7

Application to near-duplicate discovery

This chapter focuses on the application of seeded discovery to the so-called near-duplicate discovery task. While similar in principle to word discovery, this task presents specificities and challenges that deserve a special care. Not by accident, the two tasks have often been approached by different research communities (speech and multimedia), because of different sought patterns and targeted applications.

As in the previous chapter, we start by formally defining the problem at stake, before introducing necessary information, like the data set used and the performance measures adopted to provide a quantitative assessment of experimental results.

Next, a number of modifications of the baseline architecture will be presented to deal with additional issues, deriving from large-scale and match extension problems. Given the large size of the streams and the large period of occurrence of motifs, issues arise forcing the adoption of strategies to speed up the computation. Namely:

- To cope with the increased search buffer length, a fast, approximate matching technique based on downsampling of features sequences is employed.
- two faster library search strategies, Nearest Neighbor path and Nearest Neighbor model, will be introduced to deal with the growing size of the library at run time.

In addition, we describe a method to recover motif occurrences in their length by merging overlapping occurrences, to deal with fragmentation of motifs into *submotifs*, as a consequence of path extension failure.

In the following the experimental part is illustrated according to the evaluation protocol, as well as a thorough discussion on the type of motifs discovered.

7.1 Definition of the task

Near-duplicate discovery is the task of discovering and collecting in audio streams occurrences of patterns yielding a limited variability. The term *near-duplicate* is frequently used in video processing domains, where it designates video frames that are similar or nearly duplicate of each other, but appear differently due to variations introduced during acquisition time, lens setting, lighting condition or editing operation. This nomenclature is employed in our context, to highlight the difference with word discovery which mainly resides in the low amount of intra motif variability admitted. Within this class of motifs are included, for instance, several types of sound patterns: *signalling* patterns that repeat in radio or TV streams, like station call signs and signatures or applauses in TV shows and situation comedies, then advertisements, songs, even entire movies or shows, as long as they repeat. This type of audio data is often referred to as *composite*, as it presents a sequence of sounds of different nature that are mixed or follow each other (speech, music, applauses, environmental sounds ecc...).

Variability. Sources of variability can be identified in the different signal-to-noise ratio (SNR) the signal can experience during broadcast at different times of the day (or the week), or variations introduced during acquisition at the receiver. These factors do not impact signal variation so strongly as those factors responsible for the variability of the speech signal.

Motif length. Besides their limited variability, these patterns are longer than words (or small multi-word phrases) in speech. In fact their duration may vary within a wide range of values, from a few seconds, in the case of applauses or short jingles and station call signs, up to several minutes (songs), and even hours (movies, shows). It should be clear that, given the increased length and reduced variability of the sought patterns, the task of recognizing similarities is greatly simplified than in word discovery, where indeed represents the main obstacle towards building an operative computational framework.

Occurrence period. On the other hand, while the average occurrence period of repeating words is of the order of seconds or minutes in many realistic scenarios (news shows, academic lectures, conversational meetings), near duplicate patterns are expected to repeat in the span of hours (think of songs frequency of play in the broadcast schedule of a radio channel, for example). While limited variability makes similarity detection easier, the increased search space related to motif repetitiveness makes it necessary to adopt strategies to efficiently deal with large scale issues.

Main challenges posed by the task will be more clear when the test data will be illustrated and the results of the experiments presented.

7.2 Test data

The test data where experiments have been performed, comprises six 24h radio streams, sampled at 11,025 kHz and recorded from three French radio channels on March 15 and 16, 2010. The data has been provided by Yacast[1], partner within the Quaero project[2]. Together with the audio, annotations on the content are provided by enumerating the songs played that day. For each song, the following properties are specified:

- a numerical identifier
- the title of the song
- the name of the artist (or musical band)
- the record label
- the musical genre
- the day of broadcast
- starting time in hour-minute-second
- ending time in hour-minute-second

There are 1,742 songs annotated, yielding an average duration of 189 seconds. The distribution of songs' durations is reported in the histogram of Figure 7.1, from

[1] http://www.yacast.fr/fr/index.html
[2] http://www.quaero.org

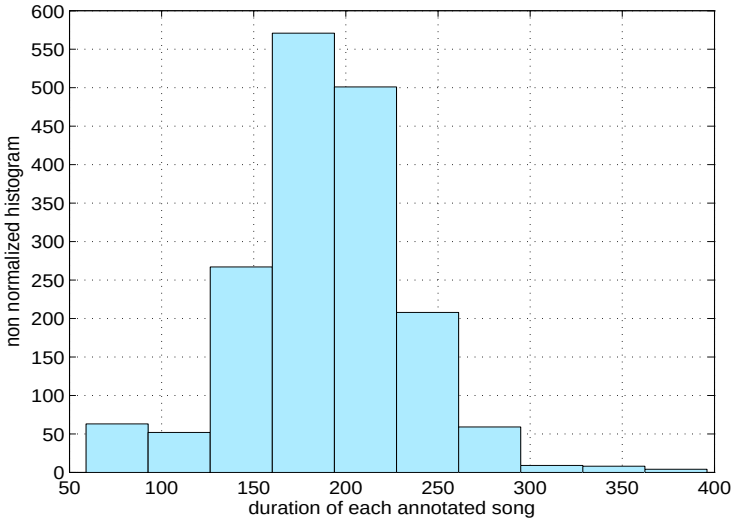

Figure 7.1: Histogram of songs' durations (in seconds) in the six 24h radio stream

which it can be observed that the most populated bin, with lengths in the range [145, 210] seconds, counts 571 occurrences. The shortest song has a duration of 59 seconds, the longest one of 396 seconds. Among these songs, 208 are repeating, which means they occur twice at least. The number of occurrences for each motif is detailed by the bar plot of Figure 7.3: the most repeating song (*Replay* by *Iyaz*) occur 14 times in a day, the average number of repetitions for each repeating song being 3.36. A crucial parameter in a motif discovery task, and even more in our computational system, concerns the occurrence period of a motif, that is the average time separation between consecutive occurrences. This value is of great relevance as it impacts our choice of the search buffer length, that is the search space assumed to include at least two occurrences of each motif. In Fig. 7.2 it is shown the histogram of the time intervals occurring between consecutive instances of each repeating song. The average value of this parameter, computed over all occurrences of all repeating songs, amounts to 5 hours and 41 minutes. From Fig. 7.2 it can be observed that repetitions may occur as soon as about 1 hour (35 minutes) and as late as almost 23 hours. It follows that to at least guarantee the possibility of retrieving all repeating songs, a search buffer must be set equal or longer than the largest time separation observed, which corresponds to almost a day of radio stream.

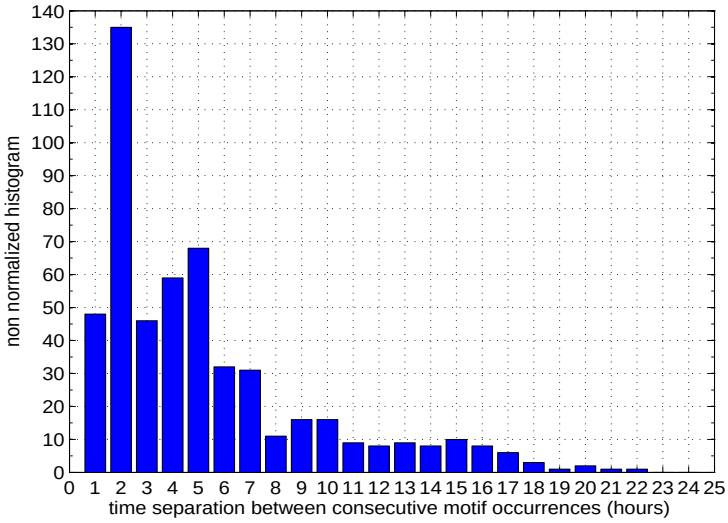

Figure 7.2: Time separation between consecutive occurrences (in hours) for all repeating songs annotated.

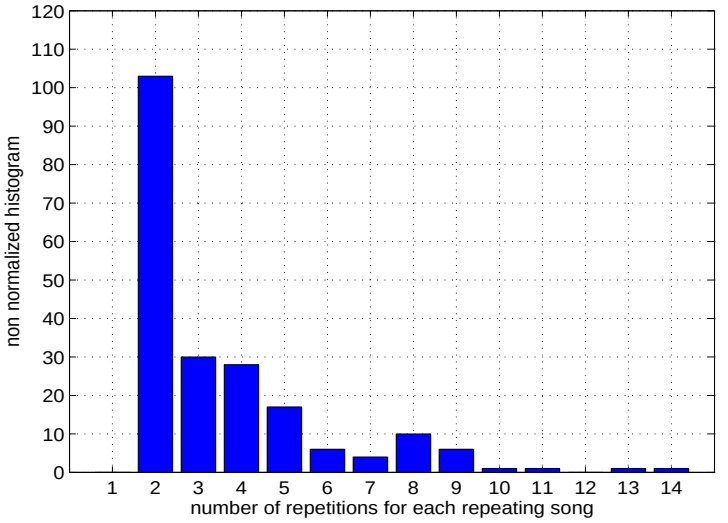

Figure 7.3: Number of repetitions for each repeating song annotated.

7.3 Performance measures

As done for word discovery, the quantitative evaluation of experiments for song discovery is accomplished by defining appropriate measures of precision and recall. The end goal is to quantify the capability of discovering real repetitions of audio segments (measured by precision) and, for each motif, as many occurrences as possible (as measured by recall). Much like in word discovery where we relied on phonetic alignments included in the ESTER corpus, we use the annotations described in the previous section to gather these statistics.

These numbers are necessarily computed over segments belonging to annotated regions of the stream, for which the evaluation process can be automatized. The remaining occurrences should be evaluated single-handidly by listening to each of them, to check whether or not they are effectively occurrences of a same motif. Being a tedious and time-consuming operation, we limit the evaluation to annotated segments, occasionally sampling the non-annotated ones to check if the algorithm has discovered other correct repetitions.

In the annotations provided, for each song only time endpoints are provided, but no information on the internal structure of the piece, *i.e.* indication of the instrumental part only, parts with speech and music mixed together, location of the refrains, chorus, etc. If two segments are recognized as belonging to the same song (whether it is the very same occurrence, or different plays of the same song), it is not possible to determine, based on the annotations only, if they also refer to the same portion of song or not. Again, since the rigorous evaluation by direct listening is not feasible, we decide to deem similarity of occurrences according to the numerical identifiers of the song they belong to: if matching, they are considered true hits. Then, for a motif i, a precision P_i is computed by a) collecting the identifiers of each occurrence b) detecting the most frequent one and c) computing the ratio between its occurrence count and the number of occurrences found for that motif.

For clarity, if $id_{\max}$ indicates the occurrence count of the most frequent identifier for a motif of cardinality N_i, then:

$$P_i = \frac{id_{\max}}{N_i} \tag{7.1}$$

and the global precision is then obtained by averaging this measure over all motifs discovered. It should be clear that the song labeled by the $id_{\max}$ identifier plays the

same role as the median occurrence defined in the word discovery evaluation protocol. It is also worth specifying that, whenever disjoint segments of the same songs are collected as occurrences, they are only counted as one for precision (and recall) computation. This is because, even though they are motifs themselves (*submotifs* indeed), we aim at evaluating motifs at the song level, according to the given annotations.

As far as recall is concerned, the most frequent song is used as representative of a given motif. We then compute the ratio between the number of occurrences collected and the number of occurrences M_i in the data, retrievable from the annotations. In short,

$$R_i = \frac{id_{\max}}{M_i} \tag{7.2}$$

and the total recall R is the average of the single recall over all motifs. Differently from the word discovery case, the repeating songs are known in advance thanks to the annotations, so that we can include the undiscovered songs in the total recall computation, each contributing with a zero recall.

Besides precision and recall, an additional measure of the performance will be given by the motif *fragmentation* value, which provides an indication of how much fragmented a motif discovered is, that is whether or not a song is retrieved by its entire length, or by portions of it. Depending on the application, it might be sufficient to discover repetitions of songs from simple excerpts, or, by contrast, from the pieces of music in their entire duration.

Before presenting the results of the actual experiments, some approximate methods will be introduced, necessary to speed up the algorithm and deal with large scale issues.

7.4 Modifications of the baseline architecture

In this section, we describe the integration of additional features into the baseline computation architecture as introduced in Chapter 5. The nature of the targeted patterns (*i.e.* songs of several minutes), the size of the data sets (a 24h continuous stream), the way the repetitions occur (after several hours), are all factors that influence performance, both in terms of computation time, and quality of repetitions found.

The need for setting a large search buffer, due to large repetition intervals, impact significantly the number of operations, and thus the computation time, required to

process the data.
Concerning the quality of motifs found, a fragmentation into submotifs will be observed, related to path extension failure during song discovery and reconstruction.

The section is then structured in two subsections. In the first subsection, we analyze large scale problems deriving from the processing of days of broadcast data. This problem can be categorized, in turn, in two main subproblems, to relate respectively to the search buffer comparison and library search modules. As far as the first subproblem is concerned, computation speed up is achieved through the use of *low resolution* versions of acoustic patterns based on downsampling, to enable a fast SLNDTW followed by a similar validation in the full resolution domain. With regard to library search, two strategies for searching possible repetitions will be illustrated, respectively called Nearest Neighbor Path and Nearest Neighbor Model, both taking advantage of the relevant speed up achievable by downsampling. The second subsection is instead dedicated to illustrating the problem of the fragmentation of motifs into shorter submotifs; a partial countermeasure will be proposed consisting in the simple merging of occurrences that overlap in time.

7.4.1 Techniques for handling large data sets

Factors influencing the processing time for each seed block are several:

- The length of the seed block.
- The length of the search buffer.
- The size of the library.
- The value of the threshold: increasing the threshold, increases the number of paths to evaluate (the candidate motifs).

The search buffer length becomes a critical parameter when set to several hours, or even an entire day, to account for occurrences distant in time. To give an idea of how impactful can be such factor, we report on a demonstrative experiment on a day of broadcast radio data.

Using a seed block of 10 seconds and a search buffer as long as the entire file, ten iterations of the algorithm were computed (that is, ten seed block were searched in a 24h search buffer, the library being empty). In average, the time required for each

seed block was 100 times the seed length. This amounts to saying that the entire file would need more than three months to be processed (disregarding additional issues deriving by the growing size of the library at run time).

An attempt to partially overcome the problem was done by resorting to vector quantization techniques, in order to reduce the burden of computing the local distances between all frames of seed block and search buffer. In vector quantization, each spectral vector is mapped into a codebook of finite size, trained properly before run time. Since the distances between codebook's *centroids* are pre-computed and stored into a look-up table, computing local distances reduces to accessing the right look-up table entry. Despite a non-negligible gain earned by applying this method, the performance were still observed to be too slow.

The key idea for successfully coping with such a long search buffer, is performing the computation on a coarse version of the audio segments, rather than in speeding up the computations in the full dimension domain. We will see in the following a practical example of this approach by sequence downsampling.

7.4.1.1 Dealing with large search buffers: Downsampling

Sequence downsampling is a simple technique for producing a low resolution representation of a time series by sampling the original sequence every k frames, k being the downsampling factor. Formally, the low resolution sequence $\hat{S}$ is computed from the full resolution sequence S as $\hat{S}(x) = S(k \cdot x)$.

The speedup in computing the distance matrix in DTW, is quadratic in the downsampling factor, if both the compared sequences are downsampled. That means $k = 10, 20, 30$ provides a $100, 400, 900$-fold gain in computation time with respect to the distance matrix computation in the full resolution domain. Thinking of the number reported in the previous section (a seed block processed in a time span 100 times its duration), the use of this approximate technique would be extremely advantageous, if performance would be demonstrated sufficiently accurate. In general, accuracy strongly depends on the task and targeted pattern, and the appropriate downsampling factor (if any), is to be determined empirically from experimental evaluation.

To this end a small scale experiment of *audio segment spotting* was conducted, consisting in searching by SLNDTW a fragment of a song into a search buffer of about 1 hour, where a repetition occurs. The experiment was performed by varying

Table 7.1: Repetition spotting: for each downsampling factor k and query length L_q, the rank of the correct path and the ratio between CPU time and query length is indicated.

	k=2		k=4		k=5		k=10		k=20	
	rank	$\frac{\text{CPU}}{L_q}$	rank	$\frac{\text{CPU}}{L_q}$	rank	$\frac{\text{CPU}}{L_q}$	rank	$\frac{\text{CPU}}{L_q}$	rank	$\frac{\text{CPU}}{L_q}$
L_q=2s	1^{st}	2.39	1^{st}	0.63	1^{st}	0.42	2^{nd}	0.12	6^{th}	0.03
L_q=4s	1^{st}	2.35	1^{st}	0.59	1^{st}	0.39	8^{th}	0.10	6^{th}	0.03
L_q =10s	1^{st}	2.45	1^{st}	0.6	1^{st}	0.38	1^{st}	0.10	1^{st}	0.03

the fragment duration (2,4 and 10 seconds) and the downsampling factor k (2, 4, 5, 10 and 20).

After SLNDTW computation, the paths computed are ranked according to the respective DTW score. We define the N-best ranked paths as the *N-best paths*, regardless of any similarity threshold. The goal of the experiment is to determine whether the true matching path, that is the path mapping the query and its repetition, is present among the N-bests, and its rank among them (if present). The hope is that of achieving a significant speedup by means of downsampling, while keeping the matching path highly ranked among the computed paths. The reason for considering only the N-best paths as well as the rank of the matching path will be explained in the next paragraph, when describing the integration of the technique in the motif discovery system.

The results of the experiment are summarized in Table 7.1. For each downsampling factor k and query length L_q, two kinds of output are indicated: the rank of the correct path among the N-bests computed (N=20) and the ratio $\frac{\text{CPU}}{L_q}$ between the time needed to perform the computation and the query length. The first measures accuracy, the second quantifies the speedup achieved. With the exception of $k = 2$, it can be observed that $\frac{\text{CPU}}{L_q} < 1$, that is the time required to accomplish the task is lower than the query length. On the other hand, the correct matching path is always present among the 20 highest ranked. For $k = 2, 4, 5$ the correct path is ranked first, for all query lengths, the lowest rank being the eighth position for $L_q = 4$ seconds and $k = 10$. The results of this small scale experiment encourages the application of the method to motif discovery, as acceptable accuracy is obtained for a fragment as short as 2 seconds, and a downsampling factor as large as 20.

7.4.1.2 Integration of downsampling in seeded discovery

Rather than performing the complete discovery in the low resolution domain, comparison of downsampled sequences is used as a fast, approximate pattern matching technique for selecting motif candidates to be validated in the full resolution domain. Pattern matching in the low and full resolution space can be regarded respectively as the similarity detection and similarity score subtasks, according to the modular decomposition proposed in Chapter 3. This decomposition is highlighted in Fig. 7.4, where the modules comprising the two different stages are detailed. More specifically:

1. two sequences enter the similarity detection module where they are downsampled and compared by SLNDTW (possibly using a different value of spectral threshold to account for feature coarsening).

2. Among all computed paths, we consider only the N-best candidate motifs. That means if there are M candidate motifs (M paths yielding a score below the spectral threshold set for the low resolution comparison), only N of them are evaluated, assuming N<M. Of course, if M>N, all M paths are retained and evaluated until a match is found. The presence of a match is determined in the full resolution domain, as audio segment representation is more accurate here. More specifically, the candidate motifs outputted by the previous comparison, are projected into the full resolution space. The corresponding occurrences undergo the SLNDTW + path extension procedure, as for the baseline system.

It is interesting to note that SLNDTW in the full resolution domain also generates a set of candidate motif paths. Here, we do not pose any limit on the number of candidate motifs to evaluate, that is we do not just consider the N-bests. The reason is that, in practice, the segments undergoing the full resolution comparison (the ones entering the similarity score box in Fig. 7.4) are of comparable length, unlike in the low resolution domain, where the whole search buffer is analyzed. The gain is indeed attained by only reserving the computationally heavy full-resolution comparison to short segments identified by low-resolution comparison, to which is left the demanding task of dealing with the (very long) search buffer.

It is worth remarking that the speedup is achieved not only by dimensionality reduction in the search buffer comparison, but also by limiting the validation to the short list of the N-bests. If all low-resolution candidate motifs are to be evaluated

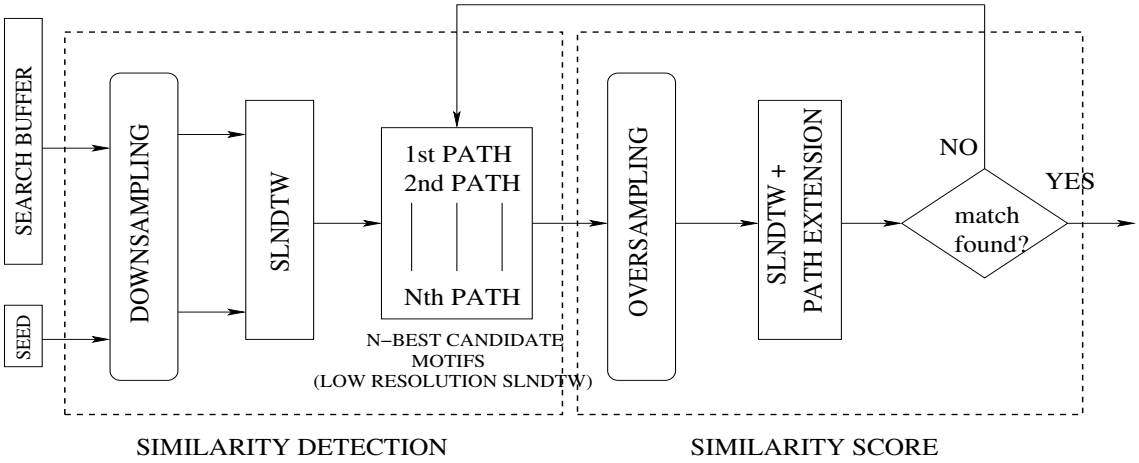

Figure 7.4: Diagram depicting the comparison in the downsampled and full domain, representing respectively the similarity detection and score subtask. The seed-search buffer comparison is performed only in the low resolution domain. In the full resolution domain only a handful of patterns (at most N pairs of candidate motifs) undergo SLNDTW and path extension procedure.

in the full resolution domain, the advantage of using fast pattern matching may be easily nullified.

We will see next an additional trick to further speed up the computation in the library.

7.4.1.3 Speeding up library search

Using a coarse audio representation and limiting the evaluation to the N-best paths, is greatly beneficial to performance in terms of computation. However the speedup concerns mainly the search buffer stage, as issues arising from the growing size of the library are still unsolved. Not only, if the comparison between a seed and a motif in library triggers (possibly) N comparisons in both resolution domains, the computation becomes even more demanding than in the baseline system (at least, in the library search). To this end, we propose two straightforward modifications of the library search procedure that we describe here:

- for each SLNDTW-based comparison between a seed and a model in the low resolution domain, the N best paths list is reduced to just the *nearest neighbor path*, that is N=1. Accordingly, each approximate comparison triggers at most

one full resolution comparison. It can be argued that, while scanning the library, each motif model can generate both a low and full resolution computation, while in the baseline system only a full resolution comparison is performed. But a) for certain values of downsampling factor, low resolution computations are extremely faster than the full resolution ones, thus almost negligible and b) in realistic situations, most low resolution comparisons do not trigger a full resolution comparison, as their DTW score is above the fixed threshold.

- each seed block is compared in the coarse domain with all motifs in the library, regardless of the presence of a possible match. Afterwards, the comparison in the full domain is performed only between the seed block and the *nearest neighbor model*, if matching. As a consequence, only a full resolution comparison for each library scan is performed at most. While this strategy forces a complete library scan for each seed, it limits the full resolution comparison to just one pair of candidate motifs. This second strategy is therefore expected to be much faster than the first one proposed.

These strategies are implemented and will be shown to be successful in allowing the algorithm to deal with large scale issues in reasonable time. Before, a practical solution to limit motif fragmentation into separated portions will be reported.

7.4.2 Recovery of motifs from overlapping segments

Several experiments have shown a recurrent failure of the path extension procedure when comparing occurrences of a same song. The failure happens as the extended path is trapped in a local minimum of the path average distortion in the space of all possible extended paths. Because of local constraints on path computation, the deviation from the true matching path might be unrecoverable, as no admissible sequence of moves can permit to subsequently rejoin the matching path (see Figure 7.5, where a visual illustration is provided). Once the matching path is out of reach, the extended path usually stops shortly after, likely to encounter some higher distortion region. This issue has been frequently observed in song retrieval, while much more rare in word discovery, probably because a) songs are much longer, and so is the path extension procedure, which is more likely to incur in local minima and b) MFCC features might be more suited to represent speech patterns, generating more distinct low-distortion regions that confine properly the extended path. The effect of path extension failure is twofold:

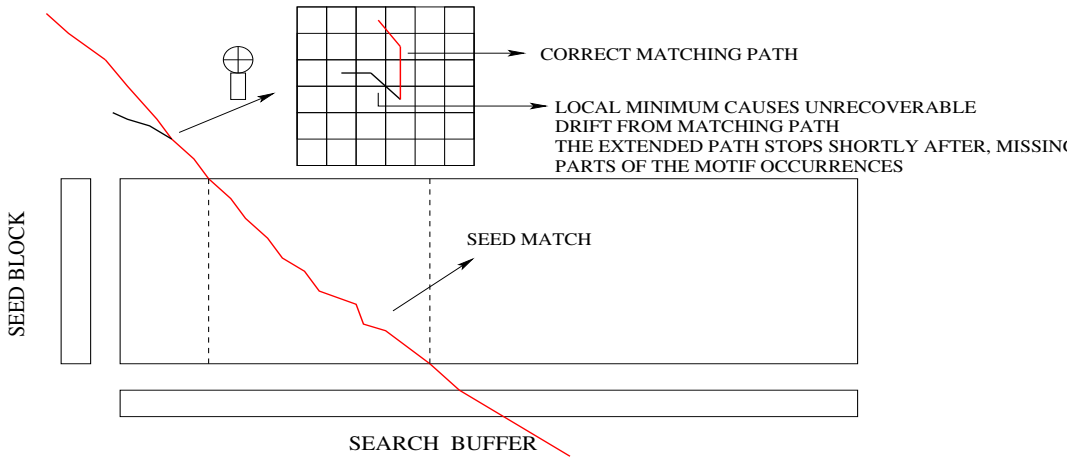

Figure 7.5: Path extension failure: the extended path is trapped in a local minimum of the path average distortion and deviates from the true matching path.

- if the occurrences identified by the prematurely stopped path, are shorter than $L_{\min}$, no match is detected at all.
- if those occurrences are instead sufficiently long, two portions of motif occurrences (*submotifs* indeed) are retrieved.

In this second case, a countermeasure can be adopted to recover the full motif occurrences and prevent motif fragmentation into submotifs. Whenever the seed following a submotif detection is matched with the subsequent portion of motif, the entire motif can be reconstructed by joining the submotifs, whenever they overlap in time (see Figure 7.6). For that to happen:

- the subsequent seed block must be deemed similar to the portion adjacent to the previous submotif occurrence.
- Path extension has to retrieve two sufficiently long submotif occurrences that stretch over the preceding ones. In this case the overlap is recognized and the merging procedure is triggered.

This countermeasure has been necessary, as our end goal is not only to correctly retrieve matching audio segments, but to reconstruct the longest possible match, which is the motif in its entire duration.

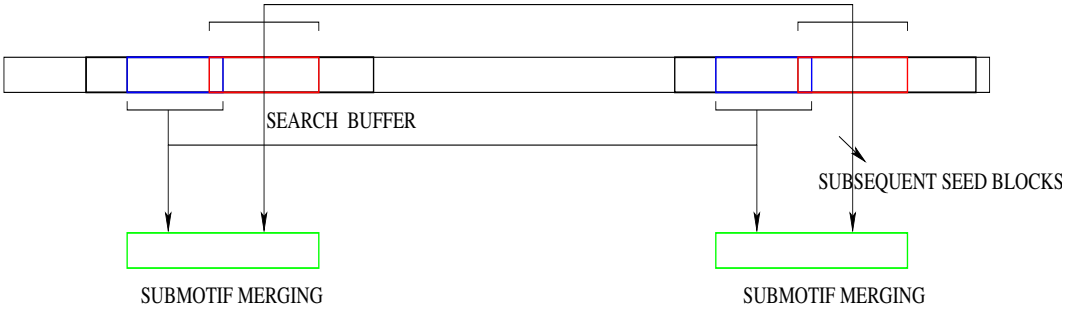

Figure 7.6: Whenever two adjacent overlapping submotifs are identified, they can be merged to overcome motif fragmentation into submotifs.

7.5 Experiments and results

After describing the additional merging feature and the various speed up techniques implemented, we are then ready to detail the experimental evaluation carried out. After a brief overview of the main parameters involved, we focus on the quantitative assessment of the performance.

7.5.1 Parameter setting

The two critical parameters in motif discovery experiments are respectively the seed and search buffer length. Their values are strictly related to the type of targeted motifs in a given task: the first is linked to the size of the sought patterns, the second to their repetitiveness. The seed block length has been set to 10 seconds, as we are mainly searching for songs of several minutes but without discarding the possibility of finding shorter repetitions like jingles or advertisements. The search buffer length has been set to at most 24 hours, basically a seed block can be searched in the entire stream. As explained before, this choice accounts for those songs observed to repeat after almost a day.

Downsampling factor. The system has been tested for different values of downsampling factor, different library search strategies, with and without merging overlapping occurrences. More specifically, two values of downsampling factor were used, namely $k = 10, 20$, as a result of the compromise between speedup and accuracy, as evidenced in the segment spotting experiment detailed previously. For both values of k the same spectral threshold has been used, higher than the threshold in the full

resolution domain (to account for desynchronization introduced by downsampling): respectively $\epsilon_{\text{DTW}} = 2.5$ in the downsampled space, and $\epsilon_{\text{DTW}} = 1.5$ in the full resolution domain.

Library search strategies. For each k, the system has been run for the two library search strategies: the one where only the nearest neighbor path generates a full resolution SLNDTW in each seed-model comparison, and the one where only the nearest neighbor motif model is considered for full resolution comparison. As for the search buffer, the list of N best paths has been set to 15 all the experiments.

Merging of overlapping occurrences. In addition, the algorithm has been run with and without merging of overlapping motif occurrences. Thus, for each 24h stream, 8 different libraries of motifs are produced.

As far as modelling is concerned, we have retained the first discovered motif occurrence as sufficiently representative of the whole class. The underlying assumption is that, being the targeted motif of limited variability, each occurrence can be indifferently assumed as representative of the motif. This was the same choice adopted by Herley in a similar task in (Herley (2006)).

7.5.2 Quantitative remarks

The results of the experiments are summarized by the statistics provided in Table 7.2. Some of these quantities have been already introduced in Section 7.3, like precision and recall, the other ones will be defined and the corresponding values commented here. Although many of these parameters tightly depend on each other, for the sake of clarity, we will account for each of them in a specific paragraph. The table can be read by rows, to compare different values of the same parameter in different configurations, or by columns, to analyze the statistics for a specific run of the algorithm. Both ways of scanning the table will be adopted, when more suited to understand the numbers.

Precision and recall. Global values of precision and recall for the six streams processed, are obtained by simply averaging the respective values computed independently for each stream.

One outstanding result that can be immediately inferred from Table 7.2 is the perfect precision gathered, equal to 1 for all runs of the algorithm. That means the

use of downsampling, even for large k, maintains a significant discrimination capability, as no false hit has even been found, despite the large size of the search space, as long as 24h of audio. As mentioned, similarity is evaluated by only comparing the numerical identifiers of the songs the collected segments belong to, without checking whether they refer indeed to the same portion of song. Every direct listening, however, has always confirmed a perfect matching; furthermore, in several cases, the matches outputted by the system have been even more precise than the annotations in detecting songs' endpoints.

Values of recall range from a minimum value of 0.67 to a maximum value of 0.83. For $k = 10$, performance are better than $k = 20$ as a result of the more accurate processing in the low resolution domain. However, recall performance do not drop significantly while doubling the downsampling factor from 10 to 20. For example, in the case of merging, recall slightly decreases from 0.7 for $k = 10$ to 0.67 for $k = 20$, both for Nearest Neighbor Model and Nearest Neighbor Path. The decrease is slightly more relevant in the no merging case, as recall drops from 0.82 to 0.70 for Nearest Neighbor Model and from 0.83 to 0.71 for Nearest Neighbor Path. One interesting observation that can be drawn is indeed the substantial identity of recall values for the two library search procedures, for a given k and merging strategy.

In general, we can conclude that recall and precision do not look greatly affected by the difference in downsampling factor between $k = 10$ and $k = 20$, when the targeted repetition is a song. This result is very attractive as it allows to achieve a significant speedup in computation time while keeping performance almost unchanged, when moving from $k = 10$ and $k = 20$. Reasons why perfect recall is not achieved are several:

1. A motif might not be discovered at all, simply because the fixed threshold has not accounted for the variability exhibited by those motif occurrences. Knowing that the repeating songs are 208 in total, it can be concluded that the algorithm has always missed one song. This can be observed in the eighth row of Table 7.2, marked by *Disc. song*, which indicates the number of discovered songs, that is the songs for which at least two occurrences (or subsegments of them) have been collected.

2. Another source of error is the path extension failure. Following a premature stop of the path extension, if the matching segments result too short to obey the minimum length condition, they are erroneously discarded.

3. A further issue is encountered when a motif occurrence is erroneously missed in the library search where the corresponding motif exists, but is then successfully found in its respective search buffer. If that happens, a new motif is added in the library, for which recall cannot be unitary, as the past occurrences, collected in the skipped motif, cannot be anymore detected (since the algorithm processes the stream in a sequential manner).

Fragmentation, motif dispersion, number of motifs found. Fragmentation measures motif segmentation into submotifs. The source of this fragmentation is due to path extension failure and has been described when merging of overlapping submotifs has been introduced. Fragmentation is measured by computing, in average, the ratio between the duration of an occurrence found and the duration of the song it belongs to (and it is reported in percentage). The third row in Table 7.2 details the value of the parameter for the different configurations.

One marked difference is visible between the systems that use merging and the ones that do not use it. For $k = 10$, fragmentation goes from 31.59 and 31.43 (for Nearest Neighbor Model and Nearest Neighbor Path respectively) to 15.3 and 15.62. For $k = 20$, it amounts to 25.7 and 25.6 with merging (for Nearest Neighbor Model and Nearest Neighbor Path respectively) against 13.8 and 15.38 without merging. This practically amounts to saying that, if a repeating song has an average duration of 3 minutes, each repetition found is fragmented in average into segments of about 45-55 seconds (respectively for $k = 10$ and $k = 20$), when merging is used, and of about 25 seconds when it is not.

Related to motif fragmentation, is also motif *dispersion*: it measures, in average, how many different motifs in library refer to the same repeating song. Motif dispersion into different motifs is again due to the path extension failure. For example, referring to Figure 7.6, if no merging is used, the repetitions marked in black are fragmented in two different motifs (the blue and red ones). If merging is used, on the other hand, they are joined into the same motif, the one indicated in green. Ideally, the desired situation is to have each motif in a library referring to a different pattern, hence the targeted value of dispersion is 1. The reported value in the implemented systems range from 2.6-3.5 when merging is used to 9-12 when not. This means that in average motif occurrences are spread into 3-4 or 9-11 different motifs, depending on the use of merging or not. This trend is clearly confirmed by the number of motifs stored in library for the different systems. By looking at the fifth row in Table 7.2,

it can be seen how the deciding factor that determines the value for this parameter is indeed the merging parameter, while remaining basically constant when varying downsampling factor or library search strategy. This parameter varies in the range [1227,1352] when merging is used and in the range [3095-3337] when it is not. Of course these numbers are obtained by summing the number of motifs independently found for each of the six streams processed.

The average number of occurrences per motif is instead quite similar for all system configurations, and amounts to about 3-4 occurrences for each motif.

Alternative definition of recall. The parameter we comment on in this paragraph is the one reported in the seventh row of Table 7.2, marked as *all occ*, which tells how many of the repeating songs annotated have been retrieved in all of their occurrences, in at least one motif in library.

This additional performance indicator can be used as a recall measure alternative to that introduced in Section 7.3 and computed for all experiments.

For clarity, suppose that a repeating song appears multiple times in different motifs; then what matters (according to this new evaluation measure) is that, in at least one of these appearances, all its occurrences are collected (even if just single excerpts for each occurrence). If that happens the discovery for that repeating song is to be considered successful and the recall unitary. In this case, other motifs where it appears can be disregarded for evaluation purposes.

According to the definition of recall we have used so far, instead, all appearances of that motif, even the ones bearing just a few of their occurrences, are accounted for evaluation purposes (which usually leads to decrease the final value, see the third source of recall degradation reported in the recall paragraph). From row 7 of Table 7.2, it can be seen that out of 208 repeating songs annotated, in the worst case 170 songs are retrieved in all of their occurrences, in the best case 197 out of 208 are detected (the best performance are observed for $k = 10$). Then, if we would measure recall by simply computing the fraction of songs for which all occurrences are found at least once, recall would be at worst 0.82 (170/208) and at best 0.95 (197/208) for the experiments performed.

This different way to measure recall has been discussed also to account for other evaluation protocols adopted in related works, so that our results can be more easily compared to them. For example, in the work of Ogle & Ellis (2007) the aim of the task is that of identifying repeated sound events in long duration personal audio

archives, including songs. Since the end goal is to facilitate browsing of massive audio collections, the user is just interested in discovering the approximate location of a song in the stream, for which retrieving single excerpts, rather than exact endpoints, is sufficient. Then a repeating song has a perfect recall if some excerpts of all its repetitions are found and grouped together (at least once).

Computation time. One very important feature for which the system is to be evaluated is the computation time required to complete the task. That is also the main reason that triggered the implementation of the different fast, approximate techniques for achieving speedup. The value of the CPU time is reported for all runs of the algorithm in the last row of the table. This is the one parameter where the difference between using $k = 10$ or 20 is most visible. For $k = 10$ the system needed at most a day exact (the stream duration indeed), and at best 17 hours, while for $k = 20$ the CPU time required ranges from about 6 hours to 12 hours. It is interesting to note that, for a fixed k, a relevant impact on computation time is implied by the specific library search strategy employed: not surprisingly the best result is achieved by using the nearest neighbor model method, as the library is completely scanned for each seed block, but only in the downsampled domain, implying at most only one comparison in the full resolution space. What is equally interesting is that the different library search strategies, while impacting the computation time, do not significantly influence the other performance indicators. It can be observed by scanning, for a fixed k and merging strategy, the columns referring to both library search methods, and noticing how similar are the corresponding values, besides CPU time.

Finally, a certain influence in computation time is due to the merging strategy, and specifically the use of online merging of overlapping occurrences is noted to impact negatively the computation time required. This is surprising as merging has the effect of reducing motif dispersion and thus the library size, which in turn should allow for a faster library scan. We argue that the source is likely to be identified in a suboptimal software implementation of the strategy. But since for all the runs of the algorithm the system has been able to accomplish the task within a day of audio at most, we can be reasonably satisfied of the speedup achieved.

Table 7.2: Statistics for the different runs of the algorithm on the 24h radio stream (see the text for detailed comment).

	k=10				k=20			
	NN model		NN path		NN model		NN path	
	m.ge	no m.ge	m.ge	no m.ge	m.ge	no merge	m.ge	no m.ge
P	1	1	1	1	1	1	1	1
R	0.70	0.82	0.70	0.83	0.67	0.70	0.67	0.71
Frag (%)	31.59	15.3	31.43	15.62	25.7	13.8	25.64	15.38
Disp	2.6	9.0	2.75	8.99	3.4	10.49	3.5	12.03
Nmotifs	1227	3095	1237	3101	1352	3322	1345	3337
Nocc/motif	3.24	3.3	3.3	3.2	3.2	3.3	3.2	3.0
all occ	182	190	191	197	170	172	172	172
Disc. songs	193	207	200	207	194	197	184	201
CPU time	19h	17.5h	23.6h	22.5h	8.6h	6.2h	12h	11h

7.6 Summary

This chapter has shown the applicability of seeded discovery to near-duplicate discovery tasks, and more specifically, to the discovery of songs within days of broadcast radio data. We have introduced some modifications of the baseline seeded discovery architecture, aiming at solving a number of large-scale issues deriving from the use of large search buffers, as imposed by the long repetition intervals of the targeted motifs. We have provided an experimental evaluation on six 24h radio streams that have clearly shown a remarkable success in retrieving most of the repeating songs, while mostly fragmented in single excerpts, rather than reconstructed in their entire duration, observing, furthermore, an excellent precision rate.

Chapter 8

Representing and accessing spoken documents: the concept of Audio Icon

This chapter focuses on illustrating the potential role played by audio motif discovery in novel information retrieval paradigms in audio data. First the problem of representing and accessing documents at the signal level is introduced, as required to properly handle massive collections of data that are available in different forms. A parallel is drawn between text mining, key frame extraction in video shot, audio summarization and audio motif discovery; we build upon these observations to finally introduce the notion of *audio icon*, as a instance of a recurrent pattern that, similarly to the small pictograms in displays, mark the presence of an event of interest.

We further elaborate on this aspect, highlighting the connection between a motif and a more specific acoustic pattern that yields *high quality* information on the audio content, as well as the utility of audio motifs for novel transcript-free mining tasks in audio.

We conclude by presenting a short demonstration on the usage of motif discovery for providing information on audio content.

8.1 The problem of representing and accessing spoken documents

The main issues underlying the efforts detailed in the thesis, revolve around a more general pair of questions that are common to a multiplicity of different fields, namely:

- How to represent a document (or a data set) in an informative way?
- How to allow for a convenient way of accessing and browsing through a document (or a data set)?

In the Internet era, given the availability of extremely massive quantity of data, in different modalities, it does not come as a surprise that a relevant effort is spent in understanding how to *decode* such information, to allow for a proper storing, indexing and browsing.

In content based multimedia indexing and retrieval, examples of tasks that focus on such problem are several:

Text mining. Text mining is the process devoted to the extraction of *high quality* information from text documents, in the sense that the targeted information is selected according to some measure of novelty, relevance and interestingness. Since the complexity of meanings, concepts and semantic relations that populate text documents, text mining is branched in many subtasks. For instance:

- text categorization: the task of assigning a document to one or more categories, according to its content. Both supervised and unsupervised methods have been proposed to deal with the task (see Joachims (1998), Sebastiani (2002)).
- text clustering: it is closely related to text categorization, and refers more specifically to the application of clustering methods for attaining automatic topic extraction, document organization, or fast information retrieval and filtering. These techniques may be used for grouping into a list of meaningful categories the thousands of search results returned by a Web search engine (an example of such software is the open source *Carrot2*[1]).
- text summarization: the task of reducing a text to a short version still containing all its important points.

[1] http://project.carrot2.org/

Key-frame extraction. Key-frame extraction is the task of abstracting a long video sequence by selection of a subset of frames, capable of retaining the visual saliency of the original shot. The targeted goals are mainly that of:

- allowing the storage of a reduced, yet relevant, version of the original sequence (possibly in the presence of limited storage capability).
- Providing a user with a summary in terms of representative still images.

Spoken document summarization. It is in fact the audio counterpart of text summarization, and it is still in its early stages, at least with respect to the well studied text summarization task. A straightforward approach, in fact, consists in resorting to ASR technologies for performing text summarization on its transcribed version. However, the possibility of performing the task directly over acoustic input, is not a far-fetched idea, and the already presented work of Zhu *et al.* (2009) is an example of such approach. We will come back later on on this work in the next section. While our work does not primarily addresses spoken document summarization, the relation is evident, as we highlight in the following.

8.2 Motif discovery and Audio Icon

A parallelism can then be drawn among the aforementioned tasks and the research subject we have studied in this thesis. The underlying end goals are very much related:

- provide, if not a summary, at least a tool to get a coarse and rapid understanding on audio/speech content.
- identify the location of coherent parts (in the sense of the acoustic similarity) to ease the access and browse through the data set.

One distinctive feature of our approach is that, to achieve this objective, no assumption is made on the semantic relevance of the targeted patterns. Then, the idea of deriving information from audio by discovery of repetitions is mainly justified by two reasons:

- recurrent patterns are *informative*, whether or not they convey meaningful semantic cues on data content. The words underlying the discovered excerpts may

refer to places, dates, characters, objects, actions. By direct listening, the user may gather different degrees of information: a) language employed, b) speakers involved c) type of data (conversational meeting, academic lecture, news show, etc...), d) the topic(s) discussed, and so on.

But even if motifs do not carry such knowledge, the identification of repetition, combined with the determination of their location, provides information on how audio content is structured and organized throughout its duration.

- finding motifs does not necessarily require a priori knowledge on the content to be performed. It is based on the recognition of similarities, and thus recurrencies, at the signal level. Assigning a score of semantic relevance to acoustic patterns require knowledge on their lexical identity that does not fit our unsupervised learning paradigm. Moreover, other than linguistic knowledge, tf-idf weights needs to be estimated for scoring and ranking the importance of words in a document.

For all these reasons, we believe that motif discovery is an interesting, preliminary tool for performing *audio mining* under the unsupervised learning framework. The term *audio mining* is used as explicit reference to *text mining* to indicate how it addresses, at least partially, the problem of extracting evidence and meaning from raw, untranscribed audio. According to these observations, we introduce the notion of *audio icon* to specifically indicate an occurrence of a motif and its use to mark parts of the data, as an audio counterpart of the pictograms in computer displays. We will see a more concrete example on the use of audio icon in a demonstrative example in Section 8.3.

From motifs to high quality audio patterns. While motifs themselves can account for audio content to a certain extent, their practical utility cannot be fully delivered if they are seen as the ultimate solution, rather than an intermediate step in audio mining tasks. As seen in the text mining case, information extraction builds upon single patterns to infer properties, meanings and structure by machine learning or clustering methods.

In the spirit of natural language processing, the natural development of our investigation consists in exploring strategies to design information retrieval (IR) systems at the acoustic level, completing the passage from transcript-based to completely audio-driven IR technologies. The ultimate goal is that of extracting automatically *high*

quality information from spoken documents, according to some criteria of saliency, redundancy, interestingness etc...(which is in fact the same role played by linguistic patterns targeted by text mining tasks)

The relation between audio motifs and these entities has not been investigated in the present work (and might very well depend on the given task and definition of high quality acoustic information), but we do believe that motif discovery can play an instrumental role towards this goal. From this regard, we can suggest ideas and cite very recent efforts that attempts to address these problems:

- **topic clustering from motifs**: this can be considered as the equivalent task of text clustering in the context of transcript-free audio mining. Conventionally, a text document is turned into a bag of words representation subject to topic-based clustering of linguistic patterns. In the bag of words model, a text is represented as a vector whose elements are frequency-based weights of the words in the text. The same model can apply in audio mining, considering a *bag of motifs* representations of the audio, each motif characterized by a specific tf-idf, estimated by discovering its occurrences in the same spoken document, and in the collection it belongs to. This is the idea at the core of the investigations detailed in (Dredze *et al.* (2010)), where preliminary experiments on topic clustering from motifs are presented. By comparison with transcript-based approaches, the authors show promising performance that encourages further study on the subject.

- **audio summarization from motifs**: in the already cited work of Zhu *et al.* (2009), audio summarization of untranscribed audio is achieved by concatenation of utterances whose importance is estimated directly at the acoustic level. Similarities at the frame and utterance level are detected relying on a slight modifications of Park's SDTW. The similarity information is used for selecting sentences that are believed to be *important*, in the sense that they are representative because frequent (show high degree of similarity with other utterances in the document), but different from other utterances already put in the summary.

- **motif importance from prosodic cues**: the idea is to exploit prosodic cues to improve semantic interpretion of acoustic patterns, which can help in assessing importance of motifs at a semantic level. There is a common agreement that features like intonation, rhythm, intensity, pausing, stress help the listeners at

different levels of speech understanding. These features have been integrated in some ASR systems to assist speech decoding. It would be interesting to see if and how they can play a useful and complementary role to infer semantic properties of motifs.

- **sentence boundary detection from motifs**: listening to the single patterns discovered may prove rather confusing in trying to understand the content of the corresponding portion of audio. These patterns are very short and outside the specific context where they appear may result rather uninformative. A simple improvement to better contextualize motifs consists in extracting, together with the repetition, the entire sentence it belongs to. Starting from the location of the pattern found, boundary sentence detection techniques can be applied and the sentence extracted may be used for a more meaningful presentation of the results.

These ideas and techniques essentially define a shift from text-based information retrieval to pure audio-driven mining approaches. It should be clear by now that the technical difficulty in discovering audio motifs, still an emerging topic in audio-speech research, is a serious drawback for the passage to audio icon and the definition of audio mining tasks. In text mining, where word transcripts are used, the document is already segmented in words, whose occurrences can be easily retrieved as the transcription of a word is unique and well defined. Conversely, in audio motif discovery, motifs are to be extracted from unsegmented, continuous data and the various instances of each motif exhibit the typical variations of speech signals. It is obvious that results approaching that of traditional NLP techniques rely on the possibility of retrieving reliable motif clusters, that is yielding high precision and recall (for example, allowing for a reliable estimation, only from the occurrences collected, of the tf-idf weights of the underlying words).

We believe, for these reasons, that our work provides also a contribution in light of this novel research areas, that are currently being explored for the first time.

8.3 Audio Icon: applicative example

A practical demonstration of audio content representation by iconic sounds is given as a result of a motif discovery task performed on a series of audio clips. These clips are

thematically coherent portions of data, of the duration of a few minutes, extracted from the *France Inter* news show, broadcast on April 18th, 2003, and already used for previous experiments in word discovery.

We provide a demo by reporting in the Table 8.1 the duration of each clip, the main subject involved, a list of audio icons, as the word patterns corresponding to the audio motifs found by the algorithm, and a list of keywords extracted from the transcript by NLP methods.

8.4 Summary

In this chapter, we have discussed the role of audio motifs among more traditional techniques to access and represent collections of documents. We have drawn a parallel between tasks such as text mining and key frame extraction in videos and audio motif discovery. In this regard, we have introduced the concept of audio icon to highlight the usefulness of motif in providing informative cues on the content of audio data. We have further discussed on the link between motifs and high quality acoustic patterns, and suggested novel audio mining tasks where motif discovery can play a crucial role.

Table 8.1: Example of audio icons as found by motif discovery. For each clip, the topic, the list of keywords, and the icons found at the acoustic level are shown.

Duration	Topic	Keywords	Audio Icons
1:58	sport news on Olimpique Marseille football club	tapie, effacé, commissaire, gaza, valence, thibaud, photos, bernard, tl, supporters	photos (2) Tapie (2) tele (2)
2:22	report on presidential elections (held on april 21st, 2002)	mathevon, réalisé, espace, barzane, cité, heures, francois, tikriti, sarkozy, hollande	la cité de l'espace a Toulouse (2) poserons (2) questions (2) gauche (2) front nationale (2) vingt-et-un avril deux-mille-deux (4)
2:44	Review of drama *Déjeuner chez Wittgenstein*	dévoué, bernhard, théatre, soeur, caché, cloos, athénée, folie, rich, méchant	Hans Peter Cloos (2) Bernard (2) fou (2) famille (2) société (2)
2:40	Psychological help for people affected by cancer	cancer, oncologie, malades, atteints, ex aequo, occupé, menacées, psychopathologie, cancérologue, maladie	cancer(3) psicohoncologies (4) jscforum (2) Nicolas Albi (2) psicologue (2) malades (2)
2:54	Easter Holidays and touristic destinations	touristique, week-end, paques, tourisme, emirats, américains, réservations, enrichir, guerre, amarrage	americains (3) touristique (3) reservation (2) hotel (3) week-end de Paque (3)

Chapter 9

Conclusions and Future work

This chapter attempts at summarizing the main contributions of the investigations detailed in the manuscript, before suggesting a number of potential future developments triggered by this work.

9.1 Summary and contributions

The doctoral work described in this manuscript has focused on the construction of a computational system for discovering repetitions of audio patterns directly from the signal and in an unsupervised manner. Such patterns, referred to as motifs, include any type of audio entities that occur in audio documents: words and short multi-word phrases in speech; jingles, advertisements, applauses or songs in composite audio. A primary merit of this work has consisted in the proposition and implementation of a unique architecture capable of dealing with different discovery tasks, like word and near duplicate discovery, characterized by a very different degree of variability and length of the targeted repetitions.

Besides the ultimate end goal of discovering motifs, a central aspect in our work resides in the unsupervised methodology at the core of our research approach. This represents a departure from traditional supervised or semi-supervised learning paradigms at the foundation of ASR technologies. While these approaches have achieved a notable success in speech and audio technologies, unsupervision offers attractive advantages, as it does not require sources of acoustic and linguistic knowledge, and it is domain and language independent. Moreover, it tries to determine to what ex-

tent a machine is able to learn and understand from audio in the absence of a priori knowledge.

In practice, the final system results from a series of single contributions that we summarize here.

First, we have proposed a division of the task into separated and autonomous subtasks, namely: segmentation, feature extraction, similarity detection and score. At this regard, we have shown how alternative systems proposed fit this modular structure. Following the logical progression implied by this modularity, we have addressed each single problem to design an initial system based on:

- a sequential stream processing based on ARGOS, that exploits statistical properties of real streams to smartly reduce the search space and avoids the combinatorial complexity of naive methods.
- the use of classic MFCC features for representing speech and a variety of different sound patterns.
- several modifications of the well known dynamic time warping algorithm, with the aim of enabling the alignment of subsequences of speech patterns, to deal with the unknown word endpoints within a continuous data stream.

This system has then evolved into a baseline architecture by proposing a simple modifications of the DTW-based pattern matching technique, so as to design a general discovery paradigm, called seeded discovery. It consists in discovering motifs by first detecting repetitions of fragments that are then extended, through a match extension technique, to retrieve the final motifs in their entire length.

From a more practical point of view, we have then shown the applicability of such an architecture to real discovery tasks. The word discovery case has been initially studied and, for evaluation purposes, novel precision-recall measures have been defined at the phonetic level. Through a series of experiment, we have shown that DTW-based seeded discovery is capable of extracting repetitions of words and word-like patterns when a limited degree of variability is enforced, resulting in a substantial speaker-dependent system.

To partially mitigate these issues, we have introduced a template matching technique based on the comparison of the self similarity matrices of speech sequences.

The idea is to investigate the spatial structure of these matrices, effectively seen as gray scale images, to recognize a two dimensional pattern dependent on the acoustic phonetic identity of the underlying sequences. We have shown, in word spotting and word discovery experiment, that the joint use of DTW and SSM based comparison, is beneficial for improving robustness of the system to speech variability. The benefit consists in the possibility of admitting an increased amount of spectral intra-motif variability, to allow for the detection of the more variable occurrences of a speech motif (like the inter-speaker ones), while ensuring an acceptable level of purity of the repetitions detected.

The usefulness of the system has also been demonstrated in a near-duplicate discovery task in broadcast radio, where the end goal is to discover repetitions of songs in a series of 24h radio streams. The success of the system has been achieved by applying a number of tricks for dealing with large scale issues, allowing to perform the task in a reasonable time. The evaluation showed the capability of the algorithm of recognizing, in the absence of any false hit, almost every occurrence of those repeating songs. The found songs have, for the most, retrieved by short excerpts rather than by their entire length.

9.2 Future work

In this section, a brief overview is provided that enumerates and describes some preliminary ideas for future developments triggered by this work.

9.2.1 Speeding up library search in word discovery

The word discovery experiments detailed, have shown the relevant impact of library search on the computation time required to perform the task. For a critical size of the library, computation can be unacceptably slowed down as each seed can be potentially compared with each of the models in library. This notably limits the utility of the system in those situations that may cause the library size to grow in excess. For instance:

- to higher ϵ_{DTW} values correspond increasing values of computation time, as shown previously. While we would like to set a value of ϵ_{DTW} to allow for a specific degree of intra-motif variability (possibly determined by a desired level

of precision-recall), computation time issues may force an upper bound to ϵ_{DTW}, limiting our freedom of choice in parameter setting.

- at a fixed value of ϵ_{DTW}, the size of the library directly depends on the size of the data set. Beyond a certain, critical size of the data set, computation time might grow uncontrolled. At the present time, scalability issues strongly constrain the size of the spoken documents that can be processed in a reasonable time.

Speeding up the library search procedure is thus a crucial problem that needs to be addressed. Two possible research directions are illustrated:

- the use of the triangular inequality for reducing the number of comparisons in library (disregarding, for the moment, that this assumption is not generally true).
- The use of fingerprints of speech templates to speed up each comparison in library.

Triangular inequality for reducing the number of comparisons. The basic idea is to exploit the triangular inequality to eliminate the need for computing all DTW-distances between a seed and the motif models in library. We can introduce the idea by referring to a DTW-based isolated word recognition scenario, which is the task of recognizing a word in a vocabulary of speech templates by DTW comparison (in fact, the library search procedure in seeded discovery can easily be regarded as an instance of such task). Consider a test word x to search in a given vocabulary V. If we assume that DTW induces a metric-space structure in the set of all possible parametric representations of words, for every $y, z \in V$ we can write that:

$$D_{\text{DTW}}(x, y) + D_{\text{DTW}}(y, z) \geq D_{\text{DTW}}(x, z) \tag{9.1}$$

which is indeed the triangular inequality. From (9.1) we can write:

$$D_{\text{DTW}}(x, y) \geq D_{\text{DTW}}(y, z) - D_{\text{DTW}}(x, z) \tag{9.2}$$

This means that the right side of the inequality lower bounds $D_{\text{DTW}}(x, y)$. Suppose we are interested in knowing whether x and y match, that is $D_{\text{DTW}}(x, y) < \epsilon_{\text{DTW}}$. If $D_{\text{DTW}}(y, z) - D_{\text{DTW}}(x, z) > \epsilon_{\text{DTW}}$, from (9.2), we can infer that $D_{\text{DTW}}(x, y) < \epsilon_{\text{DTW}}$ without directly computing $D_{\text{DTW}}(x, y)$. If all distances $D_{\text{DTW}}(y, z)$ are precomputed, we can avoid computations of $D_{\text{DTW}}(x, y)$ for all those $y \in V$ such that

$D_{\text{DTW}}(y,z) - D_{\text{DTW}}(x,z) > \epsilon_{\text{DTW}}$, for which only the computation of $D_{\text{DTW}}(x,z)$ is required. According to this procedure, a certain number of comparisons can be avoided, without any risk of skipping possible matches (according to a DTW score).

There are two main issues that prevent the straightforward application of this technique:

1. dynamic time warping dissimilarity measures cannot be assumed to be a metric as they not fulfill all the required properties (and, in particular the triangular inequality).

2. in our specific library search procedure, comparisons are performed by SLNDTW. Therefore the DTW score computed, involves the seed s and a subsegment $f_s(m)$ of a motif model m, induced by the seed. Since each subsegment depends on the given seed, is not possible to precompute the distances among the various motif models.

A positive answer to the first problem can be provided, while a solution to the second is still under investigation and might be subject to future work.

Concerning the metric properties of DTW, the most comprehensive work aiming at checking the degree of satisfaction of the triangle inequality by DTW algorithms has been published in a series of papers by Vidal and his colleagues (Casacuberta *et al.* (1987); Vidal *et al.* (1985, 1988)). In this papers they proved empirical evidence of *loose* satisfaction of the inequality by DTW in real speech. This evidence led them to propose an algorithm to reduce the number of DTW computations between the test word and the prototypes in library, to finally achieve a 70% reduction of computations with many vocabularies of different characteristics.

When adapting this strategy to our specific context, the main issues rises from the use of SLNDTW. In fact, given a seed s and two motif models m_1 and m_2, the triangle inequality can be applied so as to obtain:

$$D_{\text{DTW}}(s, f_s(m_2)) \geq D_{\text{DTW}}(s, f_s(m_1)) - D_{\text{DTW}}(f_s(m_1), f_s(m_2)) \tag{9.3}$$

Suppose that $D_{\text{DTW}}(s, f_s(m_1))$ has been already computed and we would like to possibly avoid the computation $D_{\text{DTW}}(s, f_s(m_2))$. In this case, we should be able to prove that $D_{\text{DTW}}(s, f_s(m_1)) - D_{\text{DTW}}(f_s(m_1), f_s(m_2)) \geq \epsilon_{\text{DTW}}$. As mentioned already, both $f_s(m_1)$ and $f_s(m_2)$ depends on the specific seed, hence $D_{\text{DTW}}(f_s(m_1), f_s(m_2))$ cannot be precomputed and used indifferently for all seeds.

The possible countermeasure consists in finding an upper bound, as tight as possible, of $D_{\text{DTW}}(f_s(m_1), f_s(m_2))$ that is independent of the seed. An example of such upper bound is provided by:

$$\text{UB}\left(D_{\text{DTW}}(f_s(m_1), f_s(m_2)\right) = \underset{f(m_1), f(m_2)}{\arg\max} \left(D_{\text{DTW}}(f(m_1), f(m_2))\right) \tag{9.4}$$

In this expression $f(m_1)$ and $f(m_2)$ are subsegments of m_1 and m_2 that maximize the respective DTW scores. This term is an upper bound, but not tight enough. A way of improving the selection of $f(m_1)$ and $f(m_2)$ to obtain a tighter bound is to restrict the search over those subsegments of length comparable to the seed length. This is because $f_s(m_1)$ and $f_s(m_2)$ are likely to have a length comparable to the seed length, as a consequence of the local constraints in DTW that push toward diagonality of the alignment paths.

We have not comprehensive results on this, but informal experiments seem quite encouraging in revealing the tightness of this last upper bound.

Speech fingerprints to speed up computation. A related line of research that aims at speeding up the library search operation, consists in designing and implementing fingerprints of motif models to make a single comparison faster.

The technique used in Shazam is specifically designed for identification of music patterns and previous efforts have demonstrated the inapplicability to retrieval tasks in speech (Ogle & Ellis (2007)). Among alternative techniques, it might be considered the audio fingerprinting system designed in (Haitsma & Kalker (2002)). This fingerprint is constructed by concatenating the signs of the energy differences (simultaneously along the time and frequency axes) of overlapping frames containing the spectral information of the signal.

Alternative systems might be examined from the specific literature or novel designed for our specific purpose.

9.2.2 Probabilistic modelling of motifs

A fundamental aspect impacting the recognition capability of the system is represented by the type of motif modelling. This is because, rather than comparing a speech fragment to all collected instances of a motif, we only perform pattern matching with a unique template, which is hence hoped to well represent the underlying word despite the possible variations in speech. In this thesis we have investigated

the use of three modelling strategies, namely: average, median, and random occurrence. We would like to explore alternative strategies to approach performance of those *naive* systems that do not model motifs but compare all speech fragments with another one, combining the resulting scores to identify and cluster motif occurrences. These systems are clearly more performant in collecting motif occurrences, but their complexity is also quadratic with respect to the size of the data set.

The use of a model in a lexicon of speech units is at the base of ASR systems, where those models are trained on large, labeled corpora, with the aim of producing a model that can properly handle speech variations. Producing more representative and robust models is a great challenge within our framework, because we do not rely (and we do not want to) on such training material. On the other hand, the power and flexibility of hidden markov models (HMM) for acoustic modelling have proven the main responsible of major advances in speech recognition over the past two decades. Can we take advantage of all the research efforts devoted to HMM to improve motif modelling? A similar approach was taken in (Minnen *et al.* (2007)) where seed motifs are identified by extracting the subsequences located near density modes in the distribution of data in the feature space. The detected segments are then used for training HMMs. Applied to the specific case of speech sequences, natural questions arising are:

- how to select the initial seed motifs?
- how many of them do we need to obtain reliable models?
- what kind of HMM topology is to be used?
- how to adapt the seed-match extension framework when using HMM? This basically corresponds to properly modifying the classical Viterbi algorithm as we have done with DTW.
- how to update a HMM as a new occurrence is found?

These questions could be potentially answered in future work explicitly addressing the modelling issue.

9.2.3 Performance measure

An issue of utmost importance concerns the standardization of performance indicators and data set for audio motif discovery. The lack of a common, accepted evaluation framework makes interpretation and quantitative comparison of performance extremely difficult among different systems. Usually, some form of precision-recall measure can be identified in the various work, but no benchmark currently exists.

There are several open questions that need to be addressed:

- What kind of data set is the most suitable as a benchmark data set? We have relied on broadcast news shows from different channels airing the same day. The idea was to take advantage of the presence of repeating topic-specific terms uttered by different talkers. In (Park (2006)) academic lectures are used that focus on certain topics (hence characterized by many repetitions), but those are mostly speaker-specific. In other work, like (Jansen *et al.* (2010); Minnen *et al.* (2007); ten Bosch & Cranen (2007); Zhang & Glass (2010)) discovery is mostly performed on sentences of a few seconds like those in the TIMIT corpus (as in Minnen *et al.* (2007); Zhang & Glass (2010)) or a few minutes, like the single-speaker conversation sides from the Switchboard corpus used in (Jansen *et al.* (2010)). Performing discovery on a unique stream or separately over multiple utterances has a different outcome on performance and computation time.

  In general, the type of speech data (academic lecture, conversational meeting, broadcast news show), the size of the data set, the presence of a single or several talkers have a relevant impact on performance, that might results in very different results, even when the same type of performance measure are adopted.

- What kind of annotations are to be used for evaluation purposes? We have described a phonetic-level evaluation based on the availability of phonetic alignments included in the ESTER corpus. However, by only using phonetic alignments, we cannot precisely discover how many words in the corpus are effectively repeating, an information that is crucial to measure recall rates. Appropriately combining word and phonetic level evaluation might allow for a more exhaustive interpretation of the results.

- How to define proper performance indicators? Obviously, the definition of a appropriate performance measures is also related to the availability of a certain

type of annotations. Besides precision-recall, measures that quantify sensitivity to certain sources of variability (like intra and inter-speaker variability) are useful, as well as more rigorous assessment of the scalability issues arising from a given computational system.

We believe some efforts need to be done for proposing evaluation strategies that are clear and comprehensive of all aspects of a discovery system.

9.2.4 Possible Applications

We enumerate in the following a number of novel possible application where motif discovery, or the pattern matching technique we have introduced, can be potentially useful.

Consistency checking in ASR. Motif discovery can be incorporated in ASR systems as a tool for checking consistency in the output of speech recognizers. For example, a recognizer can produce different transcripts for segments that a motif discovery system has labeled as occurrences of a same motif. We might use this information as an additional source of knowledge to be integrated in the ASR system. If the output of the discovery system is sufficiently reliable, we might want to train the recognition system such that it also deems acoustic motif occurrences as characterized by the same lexical identity.

Template based speech recognition. Recent advances in speech recognition has seen a revival of the so-called *template-based* speech recognition (Wachter *et al.* (2007)). As in the HMM framework, the recognition problem is formulated according to a Bayesian paradigm, but instead of modelling speech, actual segments of speech underlining a given word (or subword unit) are matched against a test utterance. While different sources of knowledge are fused in the recognition framework, the pattern matching technique relies on the well-known DTW algorithm. Somehow a parallel can be drawn between template-based continuous speech recognition and motif discovery, and it would be interesting to see if unsupervised matching technique, as those we have introduced, can complement and thus help the recognition within such novel ASR systems.

From text independent to text dependent speaker verification. Speaker verification is the task of checking whether two speech utterances are spoken by the same talker or not. It is used to verify if a user, claiming a certain identity, is lying or not, according to his voice. Speaker verification systems fall into two main categories: text dependent and text independent. In the first case, the text pronounced by the speaker is the same, in the second case there is no assumption on what has being said. Text-dependent systems are known to outperform the other ones, as there is no lexical variation that can generate ambiguities in the authentication. Now, suppose we are able to determine, without any a priori knowledge, whether the two speech utterances share a common pattern (a word, a group of words, or the entire sentence). If we are able to directly recognize the presence of such occurrences, a text-dependent task can be performed on them, thus approaching performance of a text-dependent system.

References

Anguera, X., Macrae, R. & Oliver, N. (2010). Partial sequence matching using an unbounded dynamic time warping algorithm. In *ICASSP '10: IEEE International Conference on Acoustics, Speech and Signal Processing*. 19, 55

Bellman, R. (1957). *Dynamic programming*. Princeton University Press. 40

Biatov, K., Hesseler, W. & Koehler (2008). Audio data retrieval and recognition using model selection criterion. In *The IEEE 2nd International Conference on Signal Processing and Communication Systems*. 36

Brazma, A., Jonassen, I., Eidhammer, I. & Gilbert, I. (1998). Approaches to automatic discovery of patterns in biosequences. *Journal of Computational Biology*, **5**, 279–305. 24

Burges, C., Plastina, D., Platt, J., Renshaw, E. & Malvar., H. (2005). Using audio fingerprinting for duplicate detection and thumbnail generation. In *ICASSP '05: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 24

Casacuberta, F., Vidal, E. & Rulot, H. (1987). On the metric properties of dynamic time warping. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **35**, 1631–1633. 135

Chai, W. & Vercoe, B. (2003). Structural analysis of musical signals for indexing and thumbnailing. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*. 24

Cheng, Y.M., Ma, C. & Melnar, L. (2005). Voice-to-phoneme conversation algorithms for speaker independent voice-tag applications in embedded systems. In

ASRU '05: Proceedings of the IEEE International Workshop on Automatic Speech Recognition and Understanding. 64

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 886–893. 90, 91

Dannenberg, R. & Hu, N. (2002). Pattern discovery techniques in music audio. In *ISMIR '03: Third International Conference on Music Information Retrieval*. 24

di Martino, J. (1985). Dynamic time warping algorithms for isolated and connected word recognition. *New Systems and Architectures for Automatic Speech Recognition and Synthesis*. 43

Dredze, M., Jansen, A., Coppersmith, G. & Church, K. (2010). NLP on spoken documents without ASR. In *2010 Conference on Empirical Methods in Natural Language Processing*. 127

Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (1998). *Probabilistic models of proteins and nucleic acids*. Cambridge University Press. 24

Furui, S. (2008). Cepstral analysis technique for automatic speaker verification. *IEEE Transaction on Signal Processing*, **29**, 254–272. 69

Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.F., Mostefa, D. & Choukri, K. (2005). The ESTER evaluation campaign for the rich transcriptions of french broadcast news. In *Interspeech-Eurospeech '05*. 70

Goto, M. (2003). A chorus-section detecting method for musical audio signals. In *In ICASSP '03: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 24

Haitsma, J. & Kalker, T. (2002). A highly robust audio fingerprinting system. In *ISMIR '03: Third International Conference on Music Information Retrieval*. 136

Herley, C. (2006). ARGOS: Automatically extracting repeating objects from multimedia streams. *IEEE Transactions on Multimedia*, **8**. 2, 3, 4, 20, 23, 28, 30, 31, 32, 53, 64, 117

Hermansky, H. & Morgan, N. (1994). RASTA processing of speech. *IEEE Transaction on Speech and Audio Processing*, **16**, 578–589. 69

Hoppner, F. (2001). Learning temporal rules from state sequences. In *IJCAI Workshop on Learning from Temporal and Spatial Data*. 25

Huet, S. (2007). *Informations morpho-syntaxiques et adaptation thmatique pour ameliorer la reconnaissance de la parole*. Ph.D. thesis, Université de Rennes 1, Rennes, France. 69

Jansen, A., Church, K. & Hermansky, H. (2010). Towards spoken term discovery at scale with zero resources. In *Interspeech 2010: Proceedings of the International Conference of the Speech Communication Association*. 19, 138

Jensen, J.H., Christensen, M.G., Murthi, M. & Jensen, S.H. (2006). Evaluation of MFCC estimation techniques for music similarity. In *Proceedings of the European Signal Processing Conference, EUSIPCO*. 36

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *In Proceedings of the European Conference on Machine Learning (ECML)*. 124

Junejo, I., Dexter, E., Laptev, I. & Pérez, P. (2008). Cross-view action recognition from temporal self-similarities. In *Proc. of the 10th European Conference on Computer Vision*, 293–306. 7

Keogh, E. & Pazzani, M. (2000). Scaling up dynamic time warping for data mining applications. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 37

Keogh, E., Chakrabarthi, K., Pazzani, M. & Mehrotra, S. (????). Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*. 30

Lamel, L. & Gauvain, J.L. (2005). Alternate phone models for conversational speech. In *ICASSP '05: IEEE International Conference on Acoustics, Speech and Signal Processing*. 69

Lattner, A.D. & Herzog, O. (2004). Unsupervised learning of sequential patterns. In *ICDM 2004 Workshop on Temporal Data Mining (TDM'04)*. 25

Lecorvé, G., Gravier, G. & Sebillot, P. (2008). An unsupervised web-based topic language model adaptation method. In *ICASSP '08: IEEE International Conference on Acoustics, Speech and Signal Processing*. 69

Lin, J., Keogh, E., Lonardi, S. & Pratel, P. (2002). Finding motifs in time series. In *ACM International Conference on Knowledge Discovery and Databases*. 24, 30, 49

Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. 30

Logan, B. & Chu., S. (2000). Music summarization using key phrases. In *ICASSP 2000: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 24

Lu, L. & Hanjalic, A. (2009). Audio content discovery: an unsupervised approach. *Multimedia Content Analysis, Signals and Communication Technologies*, **8**. 20, 21, 22, 36

Marzal, A. & Vidal, E. (1993). Computation of normalized edit distances and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 926–932. 73

Minnen, D., Starner, T., Isbell, C. & Essa, I. (2006). Discovering characteristic actions from on-body sensor data. In *International Symposium on Wearable Computers*. 30

Minnen, D., Isbell, C., Essa, I. & Starner, T. (2007). Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*. 24, 49, 137, 138

Muller, M. (2007). *Information Retrieval for Music and Motion*. Springer Verlag. 36

Munich, M. & Perona, P. (1999). Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Proceedings of the IEEE International Conference on Computer Vision*. 37

Muscariello, A., Gravier, G. & Bimbot, F. (2009a). Audio keyword extraction by unsupervised discovery. In *Interspeech '09: Proceedings of the International Conference of the Speech Communication Association*. 2, 3, 5

Muscariello, A., Gravier, G. & Bimbot, F. (2009b). Variability tolerant audio motif discovery. In *MMM '09: Proceedings of the 15th International Multimedia Modeling Conference on Advances in Multimedia Modeling*, 275–286, Springer-Verlag, Berlin, Heidelberg. 2

Newman, M.E.J. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, **69**. 16

Ogle, J. & Ellis, D. (2007). Fingerprinting to identify repeated sound events in long-duration personal audio recordings. In *In proceedings of ICASSP '07: the IEEE International Conference on Acoustics, Speech and Signal Processing*. 120, 136

Park, A. (2006). *Unsupervised Pattern Discovery in Speech: Application to word acquisition and speaker segmentation*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA. 15, 19, 30, 31, 71, 82, 138

Park, A. & Glass, J. (2005). Towards unsupervised pattern discovery in speech. In *ASRU '05: Proceedings of the IEEE International Workshop on Automatic Speech Recognition and Understanding*. 15, 22

Park, A. & Glass, J. (2006). Unsupervised word acquisition from speech using pattern discovery. In *ICASSP '06: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 15

Park, A. & Glass, J. (2008). Unsupervised pattern discovery in speech. *IEEE Transaction on Acoustic, Speech and Language Processing*, **16**. 2, 15

Peeters, G., Burthe, A. & Rodet., X. (2002). Toward automatic music audio summary generation from signal analysis. In *ISMIR '02: Proceedings of International Conference on Music Information Retrieval*. 24

Rabiner, L. & Juang, B. (1993). *Fundamentals of speech recognition*. Prentice Hall. 24, 39, 40

Rasanen, O., Laine, U.K. & Altosaar, T. (2009a). A noise robust method for pattern discovery in quantized time series: the concept matrix approach. In *Interspeech '09: Proceedings of the Conference of the International Speech Communication Association*. 30

Rasanen, O., Laine, U.K. & Altosaar, T. (2009b). Self learning vector quantization for pattern discovery in speech. In *Interspeech '09: Proceedings of the Conference of the International Speech Communication Association*. 30

Saffran, J. (2002). Constraints on statistical language learning. *Journal of Memory and Language*, **47**, 172–196. 23

Saffran, J., Aslin, R. & Newport, E. (1996). Statistical learning by 8-month old infants. *Science*, **24**. 23

Sandve, G.K. & Drabløs, F. (2006). A survey of motif discovery methods in an integrated framework. *Journal of Computational Biology*, **1**. 24

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**, 1–47. 124

Seneff, S. & Wang, C. (2005). Statistical modeling of phonological rules through linguistic hierarchies. *Speech communication*, **46**, 204–216. 69

Stouten, V., Demuynck, K. & Hamme, H.V. (2007). Automatically learning the units of speech by non-negative matrix factorisation. In *Proceedings of the European Conference on Speech Communication and Technology*, 1937–1940. 18

Tanaka, Y. & Uehara, K. (2003). Discover motifs in multi-dimensional timeseries using the principal component analysis and the MDL principle. In *International Conference on Machine Learning and Data Mining*. 30

ten Bosch, L. & Cranen, B. (2007). A computational model for unsupervised word discovery. In *Interspeech '07: Proceedings of the Conference of the International Speech Communication Association*. 2, 3, 15, 16, 22, 30, 138

Vidal, E., Casacuberta, F. & Rulot, H. (1985). Is the DTW distance really a metric? An algorithm reducing the number of comparisons in isolated word recognition. *ISCA Speech Communication*, **35**, 1631–1633. 135

Vidal, E., Casacuberta, F., Benedi, J. & Lloret, M. (1988). On the verification of triangle inequality by dynamic time warping dissimilarity measures. *ISCA Speech Communication*, **4**, 67–79. 135

Vidal, E., Marzal, A. & Aibar, P. (1995). Fast computation of normalized edit distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 899–902. 73

Wachter, M.D., Matton, M., Demuynck, K., Wambacq, P., Cools, R. & Compernolle, D.V. (2007). Template-based continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Language Processing*, **15**, 1377–1390. 139

Welling, L., Ney, H. & Kanthak, S. (2002). Speaker adaptive modeling by vocal tract normalization. *IEEE Transactions on Speech and Audio Processing*, **10**, 425–426. 69

Xie, L. (2005). *Unsupervised Pattern Discovery for multimedia sequences*. Ph.D. thesis, Columbia University, New York, NY, USA. 25

Yi, B. & Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. In *In Proceedings of the 26st International Conference on Very Large Databases*. 30

Zavaliagkos, G., Schwartz, R. & Makhoul, J. (1995). Batch, incremental and instantaneous adaptation techniques for speech recognition. In *ICASSP '95: IEEE International Conference on Acoustics, Speech and Signal Processing*. 69

Zhang, Y. & Glass, J. (2010). Towards multi-speaker unsupervised speech patterns discovery. In *ICASSP 2010: IEEE International Conference on Acoustics, Speech and Signal Processing*. 16, 138

Zhu, X., Penn, G. & Rudzicz, F. (2009). Summarizing multiple spoken documents: finding evidence from untranscribed audio. In *Proceedings of the 47th annual meeting of the ACL and the 4th IJCNLP and of the AFNLP*. 22, 31, 125, 127