IrcamCorpusTools : plate-forme pour les corpus de parole

Grégory Beller — Christophe Veaux — Gilles Degottex — Nicolas Obin — Pierre Lanchantin — Xavier Rodet

IRCAM - Institut de Recherche et Coordination Acoustique Musique
1, place Igor Stravinsky, 75004 Paris
{beller, veaux, degottex, nobin, lanchantin, rodet}@ircam.fr

RÉSUMÉ. *Il existe un éventail d'outils pour la création, l'accès et la synchronisation des données d'un corpus de parole, mais ils sont rarement intégrés dans une seule et même plate-forme. Dans cet article, nous proposons IrcamCorpusTools, une plate-forme ouverte et facilement extensible pour la création, l'analyse et l'exploitation de corpus de parole. Elle permet notamment la synchronisation d'informations provenant de différentes sources ainsi que la gestion de nombreux formats. Sa capacité à prendre en compte des relations hiérarchiques et séquentielles permet l'analyse contextuelle de variables acoustiques en fonction de variables linguistiques. Elle est déjà employée pour la synthèse de la parole par sélection d'unités, les analyses prosodique et phonétique contextuelles, la modélisation de l'expressivité, ainsi que pour exploiter divers corpus de parole en français et autres langues.*

ABSTRACT. *Corpus based methods are increasingly used for speech technology applications and for the development of theoretical or computer models of spoken languages. These usages range from unit selection speech synthesis to statistical modeling of speech phenomena like prosody or expressivity. In all cases, these usages require a wide range of tools for corpus creation, labeling, symbolic and acoustic analysis, storage and query. However, if a variety of tools exists for each of these individual tasks, they are rarely integrated into a single platform made available to a large community of researchers. In this paper, we propose IrcamCorpusTools, an open and easily extensible platform for analysis, query and visualization of speech corpora. It is already used for unit selection speech synthesis, for prosody and expressivity studies, and to exploit various corpora of spoken French or other languages.*

MOTS-CLÉS : *parole, corpus, base de données, langage de requête, multimodalité.*

KEYWORDS: *speech, corpus, database, query language, multimodality.*

1. Introduction

Les méthodes à base de corpus sont désormais très largement répandues en traitement de la parole et en traitement du langage pour le développement de modèles théoriques et d'applications technologiques. Que ce soit pour vérifier des heuristiques, découvrir des tendances ou modéliser des données, l'introduction de traitements calculatoires et/ou statistiques basés sur les données des corpus a multiplié les possibilités et permis des avancées considérables dans les technologies de la parole et du langage. La traduction automatique, la lexicométrie et l'inférence de règles grammaticales en sont des exemples en traitement automatique des langues (TAL). La reconnaissance et la synthèse de parole en sont d'autres pour le traitement automatique de la parole (TAP). De plus en plus, les besoins et les questions des deux communautés TAL et TAP se rapprochent comme le montre la récente fusion du traitement de l'oral avec celle du langage naturel. De même l'utilisation de corpus annotés prosodiquement tel que le corpus LeaP[1] intéresse aussi bien la recherche en linguistique que celle en traitement de la parole. Toutefois, cette complémentarité n'est possible que par la mise en commun des corpus. C'est pourquoi les questions de représentation et de gestion des données des corpus sont centrales.

Les corpus oraux sont constitués de deux types principaux de ressources, les signaux temporels et les annotations. Les signaux temporels sont les enregistrements audio, vidéo et/ou physiologiques, ainsi que toutes les transformations s'y rapportant (fréquence fondamentale, spectrogramme...). Les annotations sont la transcription textuelle ainsi que toutes les notations ajoutées manuellement ou automatiquement (transcription phonétique, catégories grammaticales, structure du discours...). Les différents niveaux d'annotations possèdent généralement des relations hiérarchiques et/ou séquentielles et sont synchronisés avec l'axe temporel. Les outils de gestion des corpus recouvrent tout un ensemble de fonctionnalités allant de la création et de la synchronisation des ressources, aux requêtes (pouvant porter autant sur les annotations que sur les signaux temporels), en passant par le stockage et l'accès aux données. La plupart des systèmes de gestion de corpus existants ont été développés pour des corpus spécifiques et sont difficilement adaptables et extensibles (Oostdijk, 2000). Des efforts ont été faits pour faciliter l'échange de données par la conversion de formats (Gut *et al.*, 2004) ou pour dégager une représentation formelle pouvant servir d'interface commune entre les divers outils et les données (Bird *et al.*, 2000).

Cette notion d'interface entre les méthodes et les données est à la base de la plate-forme IrcamCorpusTools présentée dans cet article. Cette plate-forme utilise l'environnement de programmation Matlab/Octave afin d'être facilement extensible. Elle permet notamment la synchronisation d'informations provenant de différentes sources (vidéo, audio, symbolique...) ainsi que la gestion de nombreux formats (XML, SDIF, AVI, WAV...). Elle est munie d'un langage de requête prenant en compte les relations hiérarchiques multiples, les relations séquentielles et les contraintes acoustiques. Elle permet ainsi l'analyse contextuelle de variables acoustiques (prosodie, enveloppe

1. Learning Prosody Project : `http://leap.lili.uni-bielefeld.de`

spectrale...) en fonction de variables linguistiques (mots, groupe de sens, syntaxe...). Elle est déjà employée pour la synthèse de la parole par sélection d'unités, les analyses prosodique et phonétique contextuelles, la modélisation de l'expressivité et pour exploiter divers corpus de parole en français et autres langues.

Dans un premier temps, cet article présente les problématiques que doivent résoudre les systèmes de gestion de corpus de parole en donnant quelques exemples de plates-formes existantes. Dans un deuxième temps, il décrit comment IrcamCorpusTools apporte des solutions originales à ces problématiques. Enfin, des exemples d'exploitations sont donnés de manière à montrer les potentialités de notre plate-forme. La connaissance de Matlab/Octave peut aider à la compréhension de notre choix pour cet environnement de programmation, mais n'est en aucun cas requise pour la compréhension de cet article, qui propose d'ailleurs au lecteur, une courte présentation générale de l'environnement.

2. Systèmes de gestion et de création de corpus de parole

Depuis l'essor de la linguistique de corpus (Chafe, 1992), de nombreux corpus annotés ont été exploités par le TAL, dont des corpus oraux comme ceux recensés par LDC[2]. La nécessité de traiter une grande quantité de métadonnées linguistiques est inhérente aux problèmes posés par le TAL. Aussi, de nombreux systèmes de gestion de larges corpus sont aujourd'hui disponibles pour cette communauté (Cunningham *et al.*, 2002). Dans le domaine du TAP, le corpus TIMIT fut le premier corpus annoté à être largement diffusé. Une tendance actuelle est à l'utilisation de corpus multimodaux avec l'intégration de données visuelles, ce qui accroît encore la diversité des formats à gérer. Permettre à une communauté de chercheurs de partager et d'exploiter de tels corpus ne pose pas simplement la question de la gestion des formats, mais aussi celles de la représentation des données, du partage des outils de génération, d'accès et d'exploitation, et du langage de requêtes associé.

2.1. *Modèle de représentation des données*

Un modèle de représentation des données doit pouvoir capturer les caractéristiques importantes de celles-ci et les rendre facilement accessibles aux méthodes les traitant. Ce modèle constitue en fait une hypothèse sous-jacente sur la nature des données et sur leur structure. Il doit donc être aussi général que possible afin de pouvoir représenter différents types de structures phonologiques et permettre une grande variété de requêtes sur ses structures.

Les modèles principalement utilisés en TAL sont des structures hiérarchiques comme celles du Penn Treebank[3] qui peuvent être alignées temporellement dans le

2. Linguistic Data Consortium : `http://www.ldc.upenn.edu/Catalog/`
3. Penn Treebank : `http://www.cis.upenn.edu/treebank/home.html/`

cas des corpus oraux. Certains systèmes comme Festival (Taylor *et al.*, 2001) ou EMU (Cassidy et Harrington, 2001) vont au-delà de ces modèles en arbre unique et supportent des hiérarchies multiples, c'est-à-dire qu'un élément peut avoir des parents dans deux hiérarchies distinctes sans que ces éléments parents soient reliés entre eux. Ces représentations sont particulièrement adaptées pour les requêtes multiniveaux sur les données du corpus. D'autres approches telles que (Bird et Liberman, 2001) ou (Müller, 2005) se concentrent sur des représentations des données qui facilitent la manipulation et le partage des corpus multiniveaux. Il s'agit généralement de représentations temporelles des données qui explicitent uniquement la séquence des événements, les relations hiérarchiques étant représentées implicitement par la relation d'inclusion entre les marques temporelles. Enfin, (Gut *et al.*, 2004) exposent une méthode et des spécifications minimales permettant de convertir entre elles les différentes représentations des données utilisées par les corpus.

2.2. *Partage des données*

Afin de pouvoir partager les corpus, comme dans le cas du projet PFC[4] (Durand *et al.*, 2005), des efforts de standardisation ont été entrepris à différents niveaux. Un premier niveau de standardisation consiste à établir des conventions sur les formats de fichiers et les métadonnées décrivant leur contenu. Ainsi, le format XML[5] s'est de plus en plus imposé comme le format d'échange des annotations. Cette solution permet la compréhension des données par tous les utilisateurs, tout en leur permettant de créer de nouveaux types de données selon leurs besoins. Un second niveau consiste à standardiser le processus de génération des données elles-mêmes. Cela conduit, par exemple, à des recommandations comme celles de la Text Encoding Initiative[6] pour les annotations des corpus oraux. Certains projets, tel CHILDES (MacWhinney, 2000) pour l'analyse des situations de dialogues chez l'enfant, proposent à la fois des normes de transcription et les outils conçus pour analyser les fichiers transcrits selon ces normes.

2.3. *Partage des outils*

Des efforts ont également été entrepris pour créer des outils libres adaptés aux annotations des ressources audio et/ou vidéo des corpus comme Transcriber[7] (Barras *et al.*, 1998) ou ELAN du projet DOBES[8]. Vis-à-vis des outils pour l'annotation, des outils de visualisation et d'analyse acoustique sont disponibles et largement uti-

4. PFC : Phonologie du Français Contemporain : `http://www.projet-pfc.net/`
5. XML : eXtensible Markup Language : `http://www.w3.org/XML/`
6. Text Encoding Initiative : `http://www.tei-c.org/`
7. Transcriber : `http://trans.sourceforge.net/en/presentation.php`
8. DOBES : documentation sur les langues rares : `http://www.mpi.nl/DOBES/`

lisés, comme WaveSurfer[9] (Sjölander et Beskow, 2000) ou Praat[10] (Boersma et Weenink, 2001). Ces logiciels permettent l'analyse, la visualisation/annotation, la transformation et la synthèse de la parole. Ils sont programmables sous la forme de scripts pour Praat et sous la forme de « plugins » pour WaveSurfer. Malheureusement, le choix de TCL/TK[11] pour ces logiciels n'est pas répandu dans les communautés du traitement du signal, de la modélisation statistique, du calcul numérique ou de la gestion de base de données. Le choix d'un format propriétaire pour les données, dans le cas de Praat, réduit considérablement les possibilités de partage de ces données qui nécessitent une étape de conversion. Cela amène ces plates-formes dédiées à la phonétique à incorporer quelques méthodes statistiques et des machines d'apprentissage, bien que leur langage de programmation ne soit pas adéquat aux calculs numériques. D'ailleurs, bien qu'affichant des annotations, ces logiciels ne sont pas munis de langage de requête, ni de systèmes de gestion de base de données.

2.4. *Langage de requête*

Pour être exploitable par une large communauté d'utilisateurs, un corpus doit être muni d'un langage de requête qui soit à la fois simple et suffisamment expressif pour formuler des requêtes variées. Une liste minimale de requêtes existe pour tout système de gestion des corpus (Lai et Bird, 2004). L'outil de requête doit aussi offrir une bonne « extensibilité », c'est-à-dire pouvoir traiter de larges corpus en un temps raisonnable. On peut distinguer deux grandes familles de systèmes utilisés pour stocker et rechercher de l'information structurée, les bases de données et les langages de balisages de textes comme le XML. Des exemples de systèmes de requête basés sur XML sont le Nite XML (Gut *et al.*, 2004) ou la version initiale d'EMU (Cassidy et Harrington, 2001). Selon ces approches, les relations hiérarchiques multiples entre les données sont stockées dans une série de fichiers XML mutuellement liés. Les langages de requête comme XSLT/XPath sont naturellement adaptés à la formulation des contraintes d'ordre hiérarchique mais la syntaxe des requêtes se complique lorsqu'il s'agit d'exprimer des contraintes séquentielles. Un effort de simplification de ces requêtes est proposé par (Gut *et al.*, 2004) avec le langage NXT Search. Cependant, les systèmes basés sur le XML offrent une « extensibilité » limitée car ils nécessitent une recherche linéaire dans le système de fichiers (Cassidy et Harrington, 2001). À l'inverse, les systèmes de base de données sont capables de stocker de très grandes quantités d'information et d'effectuer des requêtes rapides sur celles-ci. Il a été montré que les requêtes sur les hiérarchies multiples peuvent être traduites en langage SQL (Cassidy et Harrington, 2001). Cependant, le modèle relationnel étant par nature moins adapté à la représentation des contraintes hiérarchiques et séquentielles que le XML, une requête donnée en XML se traduit de manière beaucoup plus complexe en SQL. Si des langages intermédiaires plus simples comme LQL ont été proposés

9. WaveSurfer : `http://www.speech.kth.se/wavesurfer/`
10. Praat : `http://www.fon.hum.uva.nl/praat/`
11. TCL/TK : `http://www.tcl.tk/`

(Nakov *et al.*, 2005), les requêtes les plus complexes ne sont pas toujours formulables selon cette approche.

2.5. *Exploitation des données*

Une fonctionnalité essentielle des plates-formes de gestion de corpus est la possibilité d'interfacer les données (éventuellement après filtrage par des requêtes) avec des outils de modélisation. Ainsi, alors que certains environnements de développement linguistique permettent de construire, de tester et de gérer des descriptions formalisées (Bilhaut et Widlöcher, 2006), d'autres se sont tournés vers les traitements statistiques (Cassidy et Harrington, 2001). L'apprentissage automatique pour les tâches de classification, de régression et d'estimation de densités de probabilités est aujourd'hui largement employé. Qu'elles soient déterministes ou probabilistes, ces méthodes nécessitent des accès directs aux données et à leurs descriptions. C'est pourquoi certains systèmes de gestion de corpus tentent de faciliter la communication entre leurs données et les machines d'apprentissage et d'inférence de règles comme c'est le cas pour le projet EMU et le projet R[12].

3. Vers une plate-forme complète

Comme nous venons de le voir, si certains outils comme Praat apportent des solutions partielles permettant l'exploitation des corpus, peu de systèmes proposent une solution complète allant de la génération des données jusqu'aux requêtes sur celles-ci. Lorsque de tels systèmes existent, ils ont le plus souvent été conçus au départ pour une application spécifique comme la synthèse de parole (Taylor *et al.*, 2001) ou l'observation de pathologies comme c'est le cas pour le projet CSL (Computerized Speech Lab). Cela comporte des limitations intrinsèques sur le type de données, sur leur représentation et donc sur leur capacité à être partagées. Ainsi, le chercheur à la frontière du TAL et du TAP est pour le moment contraint d'utiliser une batterie d'outils dédiés et basés sur plusieurs langages de programmation, l'obligeant à effectuer de nombreuses conversions de formats et interdisant toute automatisation complète d'un processus.

3.1. *Environnement Matlab/Octave*

Matlab est un environnement de programmation produit par MathWorks[13]. Octave[14] est une solution *open-source* qui vise les mêmes fonctionnalités et conserve une syntaxe de programmation identique. Matlab tout comme Octave fournit un langage et un environnement de programmation qui permettent d'automatiser des calculs

12. R Project : `http://www.r-project.org/`
13. MathWorks : `http://www.mathworks.com`
14. Octave : `http://www.gnu.org/software/octave`

numériques, d'afficher des résultats sous la forme de graphiques, de visualiser des données multimodales (audio, images, vidéo...), et de réaliser des interfaces utilisateurs sur mesure.

Le développement d'algorithmes et de logiciels prototypes en Matlab/Octave est en général beaucoup plus aisé que dans des langages compilés comme C/C++ ou Java. Ceci est dû à plusieurs propriétés :

– le langage est interprété ;

– le calcul matriciel/vectoriel facilite le traitement des corpus ;

– il dispose d'une très grande quantité de librairies (*toolboxes*) pour le traitement des données, l'apprentissage et l'optimisation ;

– il dispose de multiples primitives et d'outils graphiques interactifs ;

– de nombreux programmes libres sont disponibles sur le Web[15].

Cet environnement est propice à la recherche où la rapidité de programmation est, en premier lieu, plus importante que la rapidité d'exécution.

Des scripts permettent la mise en série des commandes et, par conséquent, l'élaboration de séquences complexes reproductibles. Des interfaces graphiques sont facilement programmables et exploitables par des utilisateurs ne désirant utiliser qu'une partie des commandes disponibles. Des fonctions optimisées en C/C++ peuvent remplacer des fonctions Matlab pour accélérer les calculs. Enfin, une commande « system » permet d'envoyer directement des commandes au système d'exploitation, ce qui permet de lancer d'autres programmes depuis Matlab/Octave. Matlab/Octave est un environnement multiplate-forme (Mac OS, Windows et les systèmes UNIX dont Linux) devenu extrêmement puissant et répandu, utilisé par un grand nombre de laboratoires de recherche.

Par l'expressivité de son langage, par la profusion des librairies déjà disponibles, par le calcul matriciel et par une popularité forte au sein de la communauté scientifique, Matlab/Octave s'est naturellement imposé comme l'environnement idéal pour accueillir la plate-forme IrcamCorpusTools.

4. La plate-forme IrcamCorpusTools

Pour répondre aux besoins spécifiques de la parole, de son traitement et de l'analyse de corpus, la plate-forme IrcamCorpusTools a été développée et est utilisée dans une grande variété d'applications. Elle s'inscrit à l'intersection de deux domaines de recherches complémentaires : la recherche linguistique et le développement de technologies vocales. Nous la présentons dans cette section en commençant par une vue générale du système et de son architecture. Puis, nous présentons deux spécificités de la plate-forme : son langage de requête qui prend simultanément en compte des

15. `http://www.mathworks.com/matlabcentral/fileexchange/`

contraintes d'ordre linguistique et des contraintes sur les signaux ; et le principe d'autodescription des données et des outils, qui permet de répondre aux problématiques de gestion et de création de corpus abordées dans la partie 2.

4.1. *Architecture de la plate-forme*

Afin de répondre à différentes demandes de recherche et de développement industriel, l'architecture d'origine (Beller et Marty, 2006) s'est naturellement orientée vers une solution extensible, modulaire et partagée par plusieurs utilisateurs/développeurs (Veaux *et al.*, 2008). Cette mutualisation des outils et des données implique une certaine modularité tout en maintenant des contraintes de standardisation qui assurent la cohérence du système. La solution choisie repose sur le principe d'autodescription des données et des outils permettant de définir une interface commune entre ces objets. Une vue générale de l'architecture de ICT est offerte par la figure 1, elle fait apparaître la couche d'interface que nous introduisons entre les données et les outils, et qui est constituée par notre environnement Matlab/Octave. Cette architecture à trois niveaux est semblable à celle proposée pour le système ATLAS (Bird et Liberman, 2001), elle permet à différentes applications externes ou internes de manipuler et d'échanger entre elles des informations sur les données du corpus.

Les différents éléments composant IrcamCorpusTools sont des instances (objets) de classes qui forment le cœur de la plate-forme. Ces classes sont représentées dans la figure 2. Elles sont décrites par la suite et se dénomment :

– la classe *descripteur* : classe dont les instances sont des données autodécrites ;

– la classe *unité* : classe dont les instances sont des unités reliant les données entre elles ;

– la classe *fichier* : classe dont les instances sont des pointeurs vers un système de fichiers ;

– la classe *analyseur* : classe dont les instances sont des analyseurs, c'est-à-dire des outils de génération, de conversion ou de manipulation des données ;

– la classe *corpus* : classe mère regroupant un ensemble de descripteurs, d'unités et de fichiers.

Chaque classe possède un ensemble de méthodes qui constituent le langage d'interface de la plate-forme. Ce langage (en anglais) a été minimisé afin de faciliter l'abord du système pour un nouvel utilisateur, et dans le but de le rendre le plus expressif possible. Nous reprenons, à présent, chacune des classes en détail.

4.2. *Descripteurs*

L'activité de la parole est intrinsèquement multimodale. La coexistence du texte, de la voix et de gestes (faciaux, articulatoires...) génère une forte hétérogénéité des

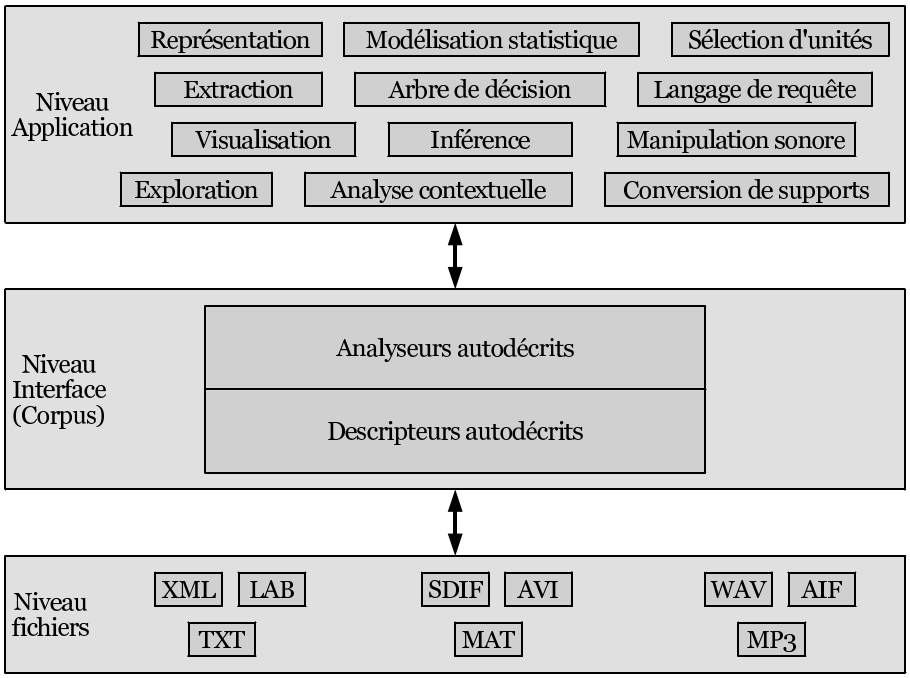

Figure 1. *Vue d'ensemble de la plate-forme IrcamCorpusTools*

données relatives à la parole. Le système doit être capable de gérer ces données de différentes natures. Voici les types de données gérées par IrcamCorpusTools.

4.2.1. *Informations de type signal*

Les signaux correspondent soit aux enregistrements provenant d'un microphone ou d'autres instruments de mesure (EGG, fMRI, ultrasons...), soit à des résultats d'analyse de ces enregistrements. Ils peuvent être unidimensionnels ou multidimensionnels. Parmi les signaux les plus courants, figurent ceux relatifs à la prosodie comme la fréquence fondamentale f_0, l'énergie, le débit de parole, le degré d'articulation mesuré à partir des formants (fréquence, amplitude, largeur de bande), et la qualité vocale (coefficient de relaxation, modèle LF, mesure du voisement), mais aussi ceux relatifs à l'enveloppe spectrale donnés par différents estimateurs (FFT, MFCC, TrueEnvelope, LPC), et représentables sous la forme de coefficients autorégressifs (AR), de paires de lignes spectrales (LSF), de pôles, ou d'aires de sections du conduit vocal (LAR). Enfin, cette liste non exhaustive peut être augmentée de signaux issus d'autres modalités comme c'est le cas par exemple pour la mesure de l'aire glottique par caméra ultrarapide.

4.2.2. *Informations de type métadonnée*

Ces informations peuvent, par exemple, servir à spécifier un contexte d'enregistrement (lieu, date, locuteur, consigne donnée, expressivité, genre de discours...). Elles comprennent les transcriptions textuelles *a priori* (parole lue) ou *a posteriori* (parole spontanée). Elles permettent de définir n'importe quelle information sous la forme de mots/symboles ou de séquences de mots.

4.2.3. *Informations de type annotation*

Comme les informations de type métadonnée, elles sont de nature textuelle. Mais elles possèdent, en plus, un temps de début et un temps de fin, permettant d'attribuer une information de type linguistique à une portion de signal. Cette sorte de donnée est cruciale pour une plate-forme de gestion de corpus de parole, puisqu'elle permet le lien entre les signaux et les catégories linguistiques, entre la physique (flux de parole continu) et le symbolique (unités de sens discrètes). Elles sont donc les pierres angulaires à la jonction entre le TAP et le TAL. Elles constituent souvent des dictionnaires clos comme c'est le cas pour les phonèmes d'une langue ou pour d'autres étiquettes phonologiques (onset, nucleus, coda...). Parmi ces informations, les segmentations phonétiques sont les plus courantes. Les annotations syntaxiques, de phénomènes prosodiques ou de mots sont autant d'étiquettes qui peuvent être placées manuellement et/ou automatiquement. Elles définissent alors des segments, aussi appelés *unités* dont la durée est variable : senone, semiphone, phone, diphone, triphone, syllabe, groupe accentuel, mot, groupe prosodique, phrase, paragraphe, discours...

4.2.4. *Informations de type statistique*

Sur l'horizon temporel de chacune des unités, les signaux continus peuvent être modélisés par des valeurs statistiques. Ces valeurs, décrivant le comportement d'un signal sur cette unité, sont appelées *valeurs caractéristiques* : moyennes arithmétique et géométrique, variance, intervalle de variation, maximum, minimum, moments d'ordre N, valeur médiane, centre de gravité, pente, courbure...

4.3. *Unités*

Les unités sont les objets permettant de relier les données entre elles. Elles sont définies pour chaque niveau d'annotation et regroupent les données symboliques ou acoustiques sur la base de la segmentation temporelle associée à ce niveau d'annotation. Les unités sont reliées entre elles par des relations de type séquentiel et/ou hiérarchique. Les relations hiérarchiques sont représentées sous la forme d'arbres (« phrase → mots → syllabes → phones », par exemple) dont les nœuds correspondent chacun à une unité. Afin de représenter des relations hiérarchiques multiples, une liste d'arbres est utilisée à la manière de Festival (Taylor *et al.*, 2001). Par exemple, les unités du niveau « phone » sont dans une relation de parenté avec celles du niveau « syllabe » et avec celles du niveau « mot » ; en revanche, les syllabes et les mots n'ont pas de relation de parenté entre eux. Ces arbres permettent de propager les marques

temporelles au sein d'une hiérarchie d'unités à partir d'un seul niveau d'annotation synchronisé avec le signal de parole (typiquement le niveau d'annotation issu de la segmentation phonétique). Inversement, à partir d'annotations indépendamment alignées, on peut construire les différentes hiérarchies entre unités, en se fondant sur l'intersection des marques temporelles. Cela permet notamment de maintenir la cohérence des diverses données relatives aux unités, tout en autorisant des interventions manuelles à tous les niveaux. À l'inverse des relations hiérarchiques, les relations séquentielles entre unités ne sont définies qu'au sein d'un même niveau d'annotation.

4.4. *Fichiers*

Nous avons choisi de stocker les différents *descripteurs* indépendamment les uns des autres afin de faciliter la mise à jour et l'échange des données du corpus (Müller, 2005). Ces fichiers reposent sur plusieurs supports dont les formats les plus répandus sont :

– LAB, XML, ASCII, TextGRID, pour les données de type *métadonnée* et *annotation* ;

– SDIF, AVI, WAV, AIFF, AU, MP3, MIDI, pour les données de type *signal* ;

– MAT (Matlab), pour les données de type *relation* et *statistique*.

En revanche, les *unités* et leurs relations sont stockées dans un fichier unique. Une fonction permet de reconstruire les unités et leurs relations lorsqu'un *descripteur* (symbolique ou acoustique) a été modifié.

4.5. *Analyseurs*

Les analyseurs regroupent toutes les méthodes de génération ou de conversion des données. On peut les enchaîner si on veut par exemple obtenir la moyenne de la fréquence fondamentale sur le groupe prosodique avoisinant une syllabe (voir l'exemple donné dans la partie 5). Certaines de ces méthodes sont dites *internes* : elles sont implémentées dans l'environnement Matlab/Octave car elles nécessitent de jongler avec les différents types de données. D'autres sont dites *externes* : elles utilisent des logiciels qui ne sont pas implémentés dans l'environnement Matlab/Octave (code exécutable, script...), mais qui peuvent être exécutés par appel depuis IrcamCorpusTools. Grâce à l'interface du système de fichiers, les données générées par un tel logiciel sont automatiquement rendues accessibles au sein de notre environnement. D'un point de vue utilisateur, le caractère interne/externe ne fait aucune différence. Dans l'exemple cité précédemment, l'utilisateur peut remplacer un estimateur interne de la fréquence fondamentale, par exemple, par celui de Praat, de WaveSurfer ou de SuperVP (Bogaards *et al.*, 2004), sans avoir à changer d'environnement.

4.6. *Corpus*

Un corpus peut être représenté comme un ensemble d'énoncés. Chacun de ces énoncés est un ensemble d'analyses. Chacune de ces analyses comportent un ou plusieurs descripteurs. Par exemple, l'analyse « audio » comporte le descripteur « forme d'onde » qui n'est autre que le signal acoustique de la phrase enregistrée. Ces analyses sont donc synchronisées au niveau de la phrase dans un corpus. Mais une synchronisation plus fine existe aussi grâce à l'ajout d'unités décrites par l'analyse « segmentation ». Les objets « Corpus » sont des interfaces avec le système de fichiers. Lorsqu'un analyseur est appliqué à un corpus, celui-ci fait appel à des fichiers d'entrée et de sortie. Le corpus stocke toute création/suppression d'un fichier, auquel il adjoint les paramètres de configuration de l'analyseur employé, ainsi que des objets descripteurs. L'objet Corpus est lui-même stocké dans un fichier XML à la racine du système de fichier, ce qui permet à plusieurs personnes d'ajouter ou de supprimer des données dans un corpus sans que cela n'entraîne de conflit. En effet, l'objet Corpus conserve au fur et à mesure l'historique des opérations effectuées sur un corpus et lui confère donc un accès multi-utilisateur.

4.7. *Langage de requête*

Certains outils de requête XML (Xpath, Xquery, NXT search) présentent une syntaxe complexe. Dans IrcamCorpusTools, nous privilégions l'expressivité du langage de requête. Une requête élémentaire est ainsi constituée :

1) du niveau dans laquelle on effectue la recherche d'unité ;

2) d'une relation séquentielle par rapport à l'unité recherchée ;

3) d'une relation hiérarchique par rapport à l'unité recherchée ;

4) d'une condition à tester sur les données numériques associées aux unités.

Ces requêtes sont rapides car elles ne s'appliquent qu'aux données préalablement stockées en mémoire vive. De plus, elles peuvent être composées afin de faire des recherches complexes prenant en compte l'interaction entre les multiples niveaux d'unités.

4.8. *Principe d'autodescription*

L'expressivité du langage de requête provient de la possibilité de mélanger des contraintes sur des données de types différents. Cela est rendu possible par le principe d'autodescription sur lequel repose IrcamCorpusTools. Chaque instance d'une classe (corpus, fichier, analyseur ou descripteur) est accompagnée de métadonnées décrivant son type, sa provenance, comment y accéder et comment la représenter. Cela permet une compréhension et une exploitation immédiate de tous les objets par tous les utilisateurs, mais aussi par le système lui-même. À l'instar du caractère interne/externe

des analyseurs, l'hétérogénéité des données est invisible à l'utilisateur qui ne possède qu'un seul lexique restreint de commandes avec lesquelles il peut rapidement se familiariser. Aucune donnée ne se « perd », car l'objet Corpus garde une trace des différentes opérations réalisées sur lui et donc, des différentes analyses ayant généré ses données. Cela permet notamment de conserver un historique de l'accès aux données. En effet, on peut toujours accéder à d'anciennes informations, même si la méthode d'accès à celles-ci a changé entre-temps. Enfin, n'importe quel utilisateur peut comprendre les données des autres et utiliser leurs analyseurs sur ses corpus, sans avoir à changer d'environnement. En résumé, le principe d'autodescription d'IrcamCorpusTools lui assure la pérennité des données, lui fournit un langage de requête expressif et lui confère la possibilité de mutualiser les données, les fichiers, les corpus et surtout, les analyseurs. La mise en commun des outils est un facteur déterminant pour le développement des recherches en TAL et en TAP, car leur complexité s'accroît rapidement.

4.9. *Exemple d'utilisation*

Nous donnons à présent un exemple d'utilisation permettant de montrer comment IrcamCorpusTools peut être utilisé. Des exemples supplémentaires sont fournis dans la partie 5. La figure 2 donne un exemple schématique d'une instance particulière de quelques objets permettant d'accéder à la moyenne de la fréquence fondamentale correspondant au phone /a/ de la 678^e phrase d'un corpus appelé Ferdinand2007.

Cette phrase est décrite par trois analyses : une segmentation en phones, une analyse phonétique, et une analyse acoustique effectuée par l'algorithme « yin » (De Cheveigné et Kawahara, 2002) qui fournit deux descripteurs continus « f_0 » (fréquence fondamentale) et « énergie » (énergie à court terme) évoluant le long de la phrase. L'analyse « segmentation » permet de définir les unités de la phrase. Une unité est décrite par son temps de début (« t_départ ») et son temps de fin (« t_fin »), son appartenance au niveau « phone » et son étiquette correspondante « /a/ ». Sa catégorie phonétique donnée par l'analyse « phonétique » est ici « voyelle orale ouverte ». L'horizon temporel couvert par cette unité permet d'accéder aux portions de signaux correspondantes (« f_0_signal » pour la fréquence fondamentale). De plus, il permet la mesure statistique de cette portion de fréquence fondamentale « f_0_moyenne » sur l'unité (valeur caractéristique).

Cette opération se réalise simplement dans notre langage par les quelques lignes suivantes.

Tout d'abord, on charge en mémoire, un corpus :

```
>> corpus = loadcorpus("Ferdinand2007");
```

Puis on instancie les objets descripteurs de la fréquence fondamentale et du phone (par la même syntaxe alors que ce sont des types différents) :

```
>> f0 = loadfeatures(corpus, 678, "f0");
```

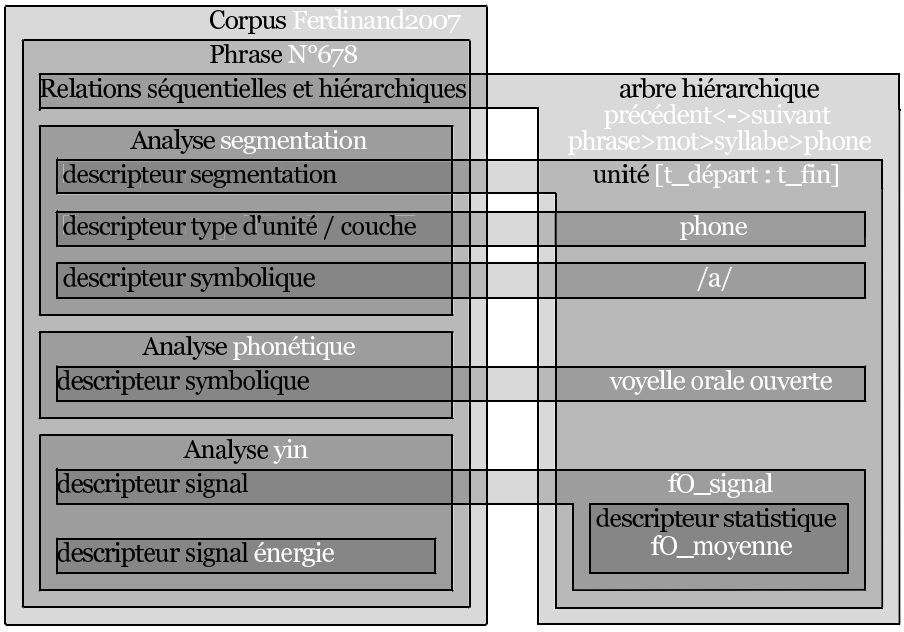

Figure 2. *Exemple d'utilisation : une instance particulière*

```
>> phones = loadfeatures(corpus, 678, "phone");
```

Enfin, on chaîne la segmentation et le calcul de la moyenne :

```
>> f0_seg = segment(f0, phones);
>> f0_mean = mean(f0_seg);
```

À cette étape, la variable résultante « f0_mean » est un tableau de plusieurs objets contenant chacun, la moyenne de la fréquence fondamentale d'un des phones de la segmentation. Afin d'accéder à la moyenne d'intérêt, c'est-à-dire à celle correspondant au phone /a/, il nous faut filtrer ces objets à partir de considérations linguistiques. Nous avons donc besoin du langage de requête.

```
>> phone_a = getunits(corpus,678,"phone",{"phoneme","is","a"});
```

Cette requête donne accès à l'ensemble des phones /a/, présents dans la phrase. La même requête sans le numéro de la phrase donne accès à tous les phones /a/ du corpus.

```
>> phones_a = getunits(corpus, "phone", {"label","is","a"});
```

À présent, nous allons enrichir cette requête de manière à étudier un phénomène de coarticulation : une voyelle /a/ précédée de la plosive /p/ et suivie de la fricative /f/. Pour cela, nous réduisons le faisceau des unités sélectionnées à partir de contraintes contextuelles, ce qui se réalise simplement en spécifiant les contraintes en chaîne dans la requête.

```
>> phones_paf = getunits(corpus, "phone",
                {"label","is","a"},
                {"prev_phone_label","is","p"},
                {"next_phone_label","is","f"});
```

Pour observer si l'effet de coarticulation partage des propriétés acoustiques similaires lorsque les contextes présentent des similarités phonétiques, nous allons maintenant élargir le contexte gauche aux plosives et le contexte droit aux fricatives. Cette requête peut se formuler de deux manières :

1) en spécifiant manuellement des listes de phonèmes :

```
>> phones_PaF = getunits(corpus, "phone", {"label","is","a"},
                {"prev_phone_label","is",{"p","t","k"}},
                {"next_phone_label","is",{"f","s","S"}});
```

2) ou alors en faisant une requête directement sur la classe phonétique des phonèmes considérés :

```
>> phones_PaF = getunits(corpus, "phone", {"label","is","a"},
                {"prev_phone_class","is","P"},
                {"next_phone_class","is","F"});
```

Par ailleurs, la représentation des données dans IrcamCorpusTools permet la description des observations sur plusieurs niveaux. Nous pouvons donc accéder, pour un niveau donné (ici, le phonème), à des informations d'un niveau parent (par exemple : la syllabe). Ce faisant, nous rajoutons à présent sur la requête précédente, le désir d'observer un phonème /a/ de même contexte mais faisant partie d'une syllabe proéminente.

```
>> phones_PaF_P = getunits(corpus, "phone", {"label","is","a"},
                  {"prev_phone_class","is","P"},
                  {"next_phone_class","is","F"},
                  {"syllable_phone_proeminence","is","P"});
```

Enfin, une particularité de notre langage, qui le rend propice aux applications TAP, est l'introduction de contraintes sur les données de type signal dans la composition des requêtes. Ainsi, l'adjonction de la contrainte {"f0_mean",">","200"} dans la requête précédente, permet d'écarter les unités dont la fréquence fondamentale est inférieure à 200 Hz :

```
>> phones_PaF_f0 = getunits(corpus, "phone", {"label","is","a"},
                   {"prev_phone_class","is","P"},
                   {"next_phone_class","is","F"},
                   {"f0_mean",">","200"});
```

5. Exploitations

Nous présentons, dans cette partie, des exemples d'exploitation d'IrcamCorpus-Tools. Certaines de ces exploitations sont reliées à la création de corpus, d'autres à la modélisation et à l'analyse, tandis que certaines sont dédiées à la manipulation du signal de parole et à sa synthèse. Si chacune d'elles est indépendante des autres, leur réunion au sein d'une même plate-forme autorise des enchaînements qui rendent automatiques des processus complexes. À titre d'exemple, la transformation de l'expressivité peut être appliquée à une phrase synthétisée ou à une phrase enregistrée dont la segmentation phonétique est automatique, tout en reposant sur des modèles appris sur d'autres corpus et ce, sans aucune intervention humaine dans le processus.

5.1. *Création de corpus*

5.1.1. *Conception de corpus*

Si l'approche de certains linguistes, qui entreprennent de soumettre leurs hypothèses théoriques à l'épreuve des grands corpus oraux, est de plus en plus répandue, c'est parce que la taille de ces corpus leur permet d'être considérés comme exhaustifs (sous certaines hypothèses) (Habert, 2000). Pour le reste, l'approche traditionnelle consiste à créer des corpus en vue de valider certaines hypothèses théoriques, prises en compte lors de la conception de ces corpus. Il en va de même pour le concepteur d'un synthétiseur de parole qui débute par une phase de conception de corpus, afin de minimiser les traitements ultérieurs.

Un ensemble d'outils de TAL a été élaboré dans le but de sélectionner des ensembles de phrases respectant certaines contraintes linguistiques. Ces ensembles sont extraits d'un large corpus textuel Corpatext[16] de plus de 37 millions de mots. L'extraction est motivée par différentes recherches de couvertures maximales sous contraintes. Pour la synthèse TTS, l'ensemble des phrases retenues doit présenter le meilleur compromis entre une taille minimale et une couverture maximale des phonèmes par rapport à des contextes donnés (phonétique, lexical, syntaxique...). Ici, une couverture maximale pourra être interprétée comme ayant au moins un candidat pour chaque contexte ou bien comme ayant une distribution des candidats reflétant une distribution naturelle (comme la distribution sur tout le corpus textuel, par exemple).

16. Corpatext : `http://www.lexique.org/public/corpatext.php`

5.1.2. *Décodage acoustico-phonétique*

Pour permettre des études en linguistique de corpus, il est nécessaire qu'un certain nombre d'étapes soient automatisées. Dans le cadre de la synthèse de parole, de modélisations prosodique et expressive, le décodage acoustico-phonétique est une étape essentielle en amont d'une chaîne de traitements linguistiques permettant de représenter la structure de la parole. Cette étape permet la segmentation d'un signal de parole en ses unités linguistiques minimales. Celles-ci sont ensuite regroupées en des unités linguistiques de dimensions supérieures (syllabes, groupes accentuels, groupes prosodiques). Une fois la conception du corpus réalisée (parole de laboratoire ou parole spontanée), les enregistrements sont automatiquement segmentés en phones à l'aide de l'analyseur *ircamAlign* (Lanchantin *et al.*, 2008). Ce dernier prend en entrée le signal de parole, sa transcription textuelle correspondante ainsi qu'un dictionnaire constitué de modèles statistiques paramétriques (Hidden Markov Models, HMM (Rabiner, 1989)) de chacun des phones en contexte, appris sur le corpus multilocuteur BREF80 (Lamel *et al.*, 1991). À partir de la transcription textuelle et du dictionnaire, un modèle statistique de la phrase est constitué prenant en compte les différentes variantes de prononciations. La meilleure séquence de phones peut alors être sélectionnée puis alignée sur le signal de parole. Finalement, afin de détecter les erreurs éventuelles et de simplifier une phase de correction manuelle, un indice de confiance est associé automatiquement à chacun des phones segmentés.

5.1.3. *Création des unités*

Le système IrcamCorpusTools offre une grande modularité dans l'étape de spécification des unités, ce qui permet d'envisager un large champ d'applications possibles en étude de la parole. Il est ainsi possible de définir arbitrairement une structure de parole (tant au niveau des unités utilisées que de leurs attributs associés) à partir de considérations particulières au domaine d'étude considéré. Cette propriété se révèle nécessaire dans l'étude des phénomènes rattachés à la parole, que ce soit pour définir des structures de la parole à partir de théories phonologiques spécifiques au sein d'une langue, pour représenter la variabilité des structures observées entre les langues ou bien alors pour définir des niveaux d'analyses supplémentaires pour des domaines d'études spécifiques (acquisition du langage, pathologie...).

À partir de la segmentation phonétique présentée précédemment, la représentation de la structure phonologique segmentale et suprasegmentale de la parole dans IrcamCorpusTools est décrite de la manière suivante : le phonème et ses attributs phonologiques, la structure syllabique (onset/rhyme (nucleus/coda)), la syllabe et ses attributs phonologiques, le groupe accentuel, le groupe prosodique et le discours (*cf.* figure 3).

Les attributs phonologiques du phonème sont :

– ses traits phonologiques (Gussenhoven et Jacobs, 2005) :

- classe majeure,
- caractéristiques laryngées,

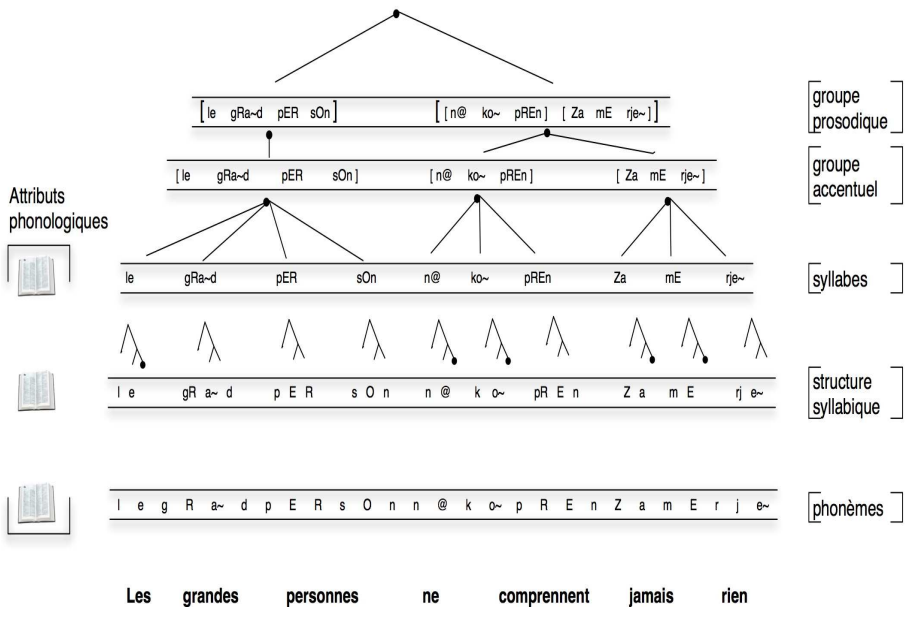

Figure 3. *Représentation de la structure phonologique en français parlé*

- caractéristiques articulatoires (continu, nasal, strident, latéral),

- caractéristiques de localisation articulatoire (labiale, coronale, dorsale, radicale) ;

– sa classe phonétique (X-SAMPA) :

- phonèmes vocaliques : voyelles (nasale/orale), semi-voyelle (glides), schwa,

- phonèmes consonantiques : consonnes liquide, fricative, occlusive (sourde/sonore), nasale.

Les attributs phonologiques de la syllabe sont :

– son type (V, CV, CVC...) ;

– sa structure (onset/rhyme (nucleus/coda)) ;

– son caractère proéminent/non proéminent.

5.1.4. *Analyse acoustique : étude de la qualité vocale*

En parallèle et indépendamment de la structure du corpus spécifiée, il est possible d'associer des analyses acoustiques sous la forme d'analyseurs. Dans le cas de l'analyse et la synthèse de l'expressivité, l'un des paramètres prosodiques importants est la qualité vocale. C'est pourquoi nous cherchons, par inversion du conduit vocal, à

estimer le débit d'air au niveau de la glotte. De nombreuses méthodes existent déjà (Vincent *et al.*, 2005 ; Henrich, 2001), mais la difficulté réside dans leurs validations. En effet, il n'existe aucun moyen de mesurer *in vivo* ce débit glottique. Cependant, à l'aide de la vidéo-endoscopie à haute vitesse, il est possible de filmer la glotte à un taux de 4 000 images/seconde. Ces images permettent d'estimer l'évolution temporelle de l'aire de la glotte (Degottex *et al.*, 2008a ; Degottex *et al.*, 2008b). Cette évolution permet l'estimation d'un débit glottique qui est alors comparé à celui obtenu par inversion du conduit vocal. Dans ce contexte, il est donc indispensable de pouvoir manipuler des informations autant visuelles qu'acoustiques s'exprimant dans une ou plusieurs dimensions. La plate-forme IrcamCorpusTools permet une visualisation synchronisée de ces différentes informations et facilite ainsi l'interprétation des données multimodales. L'ensemble des paramètres glottiques estimés sont alors calculés sur le corpus par un analyseur acoustique, et accessibles par le langage de requête.

Les étapes de construction d'un corpus et la spécification de ses différents niveaux d'analyses pertinents d'un point de vue linguistique, couplées à l'estimation de signaux relatifs à la parole, permettent un grand nombre d'études linguistiques et/ou statistiques sur des régularités au sein de ces corpus. La recherche phonologique constitue notamment une de ses multiples possibilités : stylisation de l'intonation et relations formes/fonctions (Hirst et Espesser, 1993 ; Hirst *et al.*, 2000 ; d'Alessandro et Mertens, 1995).

5.2. *Analyses linguistiques et modélisations*

5.2.1. *Caractérisation acoustique du phénomène de proéminence*

La proéminence est un phénomène prosodique majeur pour l'analyse et la modélisation de la prosodie (Rosenberg et Hirschberg, 2007 ; Avanzi *et al.*, 2008). La présente étude se place dans une approche *bottom-up* de l'analyse de ces phénomènes en trois temps : dans une première étape, des outils statistiques sont utilisés pour permettre l'émergence des corrélats acoustiques de la proéminence et permettre leur détection automatique ; dans une seconde étape, les proéminences détectées automatiquement seront utilisées pour faire émerger un ensemble de formes de la proéminence ; enfin, ces formes prosodiques seront étudiées par des linguistes pour réaliser une correspondance forme/fonction. Nous nous arrêtons ici seulement sur la première étape de cette étude : l'émergence automatique des corrélats acoustiques de la proéminence et sa détection automatique. Une modélisation statistique des corrélats acoustiques de la proéminence est rendue possible grâce à la complémentarité des possibilités suscitées d'IrcamCorpusTools et des méthodes statistiques implémentées en Matlab (Obin *et al.*, 2008c). À titre d'exemple, nous décrivons ci-dessous les étapes menant à une caractérisation acoustique de la proéminence reposant sur la hauteur f_0 moyenne des unités « syllabes », relativisée par rapport à la hauteur f_0 moyenne des syllabes adjacentes ou par rapport à la hauteur f_0 moyenne du « groupe prosodique » parent. Nous accédons aux unités syllabes de la phrase 678 du corpus Ferdinand2007, ainsi qu'à leurs f_0 moyennes comme précédemment (voir partie 4.9) :

```
>> syls = loadfeatures(corpus, 678, "syllabe");
>> f0_mean_syl = mean(segment(f0, syls));
```

Grâce aux relations entre unités, on accède au groupe prosodique parent de chaque syllabe et à leurs f_0 moyennes respectives :

```
>> prosos = getparent(corpus, syls, "prosodic");
>> f0_mean_proso = mean(segment(f0, proso));
```

Enfin, des valeurs relatives sont déterminées pour chaque syllabe, en divisant leurs hauteurs moyennes sur la hauteur moyenne de leur groupe prosodique parent respectif (la fonction gv() extrait les valeurs des objets) :

```
>> f0_mean_syl_rel = gv(f0_mean_syl) / gv(f0_mean_proso);
```

Cette étape montre comment les relations hiérarchiques entre les différentes unités de la phrase permettent à une unité d'un niveau donné d'hériter des données associées de ses « parents » ou d'agréger les données associées à ses « enfants ». La figure 4 montre le résultat de la hauteur moyenne de la syllabe relativisée par rapport aux hauteurs moyennes des syllabes adjacentes, ainsi qu'à la hauteur moyenne du groupe prosodique incluant cette syllabe.

On peut utiliser ces procédés d'analyse pour tout type de signaux et sur un corpus entier (ou sur la réunion de plusieurs corpus). Dans l'étude présentée, nous avons utilisé ces procédés pour générer une description du signal de parole sur toutes les syllabes. Cette description comprend :

– plusieurs corrélats acoustiques (fréquence fondamentale, durées, intensité, information spectrale et information spectrale perceptive) ;

– plusieurs caractéristiques de ces corrélats au niveau de la syllabe (valeur moyenne, valeur maximale...) ;

– plusieurs fenêtres temporelles contextuelles qui permettent de relativiser la valeur observée sur une syllabe donnée en fonction des valeurs observées dans son contexte (aucun, syllabes adjacentes, groupe accentuel précédent, groupe prosodique précédent).

Vis-à-vis d'une annotation de la proéminence manuelle ou découlant de cette description (automatique), il est alors possible de filtrer les syllabes présentant une proéminence grâce à la requête :

```
>> syl_pro = getunits(corpus, "syllabe", {"prominence","is","P"});
```

On récupère, de la même façon, les syllabes ne présentant pas de proéminence. La figure 5) montre cette distinction binaire dans un espace réduit de la description dont les axes sont la durée et la f_0 moyenne relativisée.

Les corrélats acoustiques de la proéminence sont appris grâce à des outils statistiques de Matlab (machines d'apprentissage) dont le but est de déterminer un ensemble

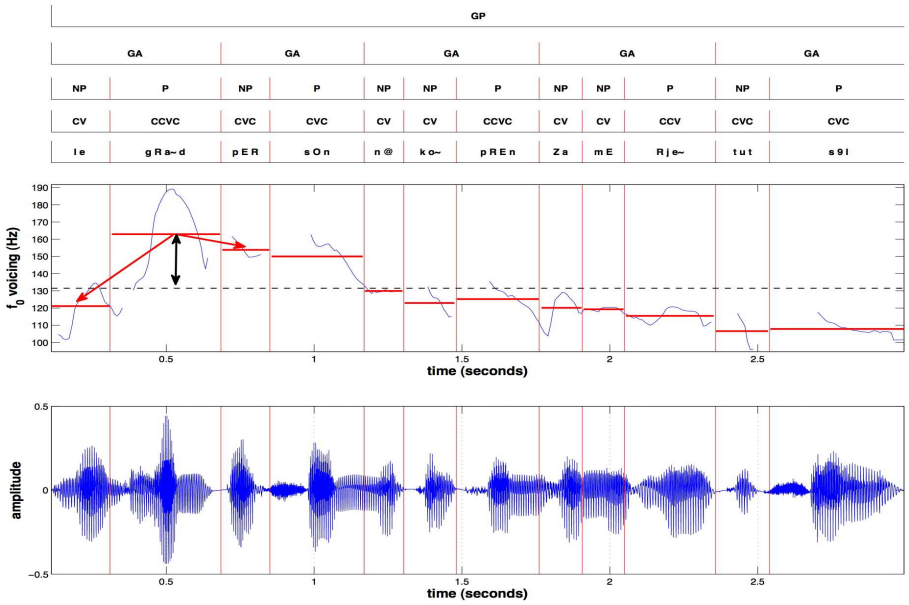

Figure 4. *Estimation de la hauteur moyenne d'une unité syllabe (trait plein) et relativisation de cette valeur par rapport aux moyennes des unités syllabes adjacentes (flèches simples) et à la moyenne du groupe prosodique (trait en pointillé et double flèche)*

ordonné des corrélats acoustiques observés sur la syllabe qui permet la meilleure discrimination entre les syllabes proéminentes et les syllabes non proéminentes.

La puissance du langage de requête d'IrcamCorpusTools a ainsi permis la caractérisation et la modélisation de la proéminence sur un corpus de voix parlée monolocuteur (Obin *et al.*, 2008c). Grâce à la facilité d'intégration d'analyseurs externes, cette méthode a été confrontée à d'autres sur des corpus de parole spontanée (Obin *et al.*, 2008a). Enfin, elle a permis la mise en place d'une méthode de caractérisation automatique des genres de discours (Obin *et al.*, 2008b).

5.2.2. *Modélisation et transformation contextuelles de l'expressivité*

L'analyse/synthèse de l'expressivité dans la parole est un nouvel enjeu pour la communauté TAP. Elle permet de rendre les systèmes de reconnaissance vocale plus robustes et d'accroître le registre des synthétiseurs TTS. De plus, elle est un outil pour les psychologues/psychanalystes qui étudient les émotions et les éventuelles pathologies qui y sont liées. Notre approche est avant tout motivée par le désir de modifier l'expressivité d'une phrase parlée, qu'elle soit synthétisée ou bien enregistrée,

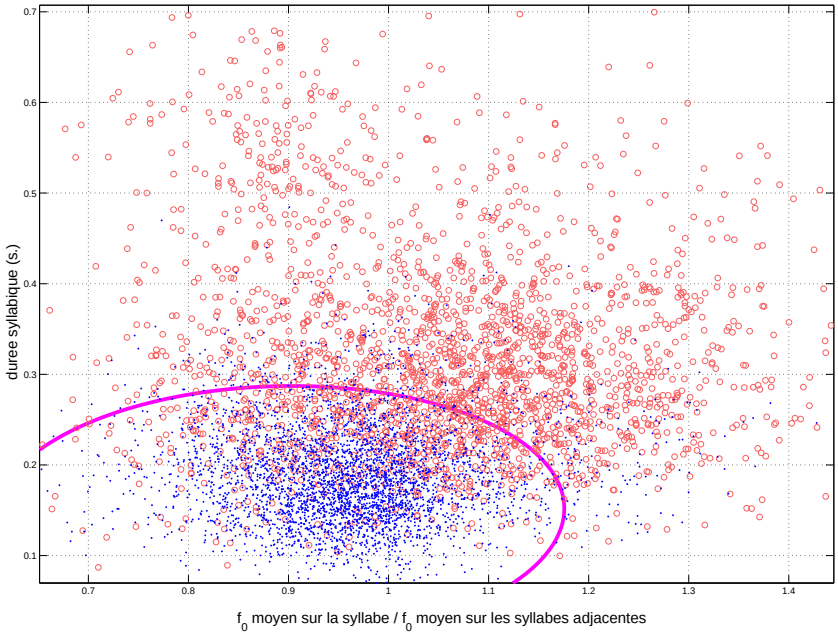

Figure 5. *Distribution des syllabes proéminentes (cercles clairs) et non proéminentes (points sombres) dans un espace prosodique choisi. Le trait plein gras représente leur courbe de séparation quadratique*

à l'instar d'un acteur. C'est pourquoi nous avons enregistré plusieurs acteurs (deux hommes et deux femmes) exprimant un même texte avec différentes expressivités et avec différents niveaux d'intensité expressive. Ces quatre corpus servent à l'établissement de modèles de jeux d'acteur. Ces modèles sont dépendants de variables contextuelles et linguistiques (Beller et Rodet, 2007) telles que le phonème ou le degré de proéminence. Par exemple, les variations acoustiques liées à l'expressivité varient selon le degré de proéminence (particulièrement pour le débit de parole (Beller *et al.*, 2006 ; Beller, 2007b)). La transformation du degré d'articulation nécessite la connaissance du contexte phonétique (Beller, 2007a ; Beller *et al.*, 2008). La capacité d'IrcamCorpusTools à gérer différents types de données y est donc pleinement exploitée. Les variables linguistiques sont utilisées par un réseau bayésien pour estimer des densités de probabilités conditionnelles des variables acoustiques relatives aux cinq dimensions de la prosodie (hauteur, débit de parole, intensité, degré d'articulation, qualité vocale) (Pfitzinger, 2006). La comparaison de ces densités amène à différents facteurs de transformation utilisés par un vocodeur de phase (Bogaards *et al.*, 2004) pour modifier l'expressivité d'une phrase neutre. Des exemples sonores sont disponibles[17].

17. `http://recherche.ircam.fr/equipes/analyse-synthese/beller`

5.3. *Synthèse de la parole*

Le langage de requête peut être utilisé de manière manuelle par une succession de lignes de commandes comme dans l'exemple précédent. Mais il peut aussi être invoqué de manière automatique à plusieurs niveaux, de manière à construire des arbres de données par de multiples décisions successives. Un synthétiseur de parole à partir du texte (TTS) a été construit de cette façon. Les procédés employés dans les systèmes TTS à base de corpus sont aujourd'hui bien connus (Hunt et Black, 1996), mais nous présentons ce système, *ircamTTS*, car c'est une application du langage de requête.

Après une phase d'apprentissage automatique impliquant de nombreuses requêtes sur les données symboliques de plusieurs niveaux (phones, diphones, syllabes, mots, groupes prosodiques...), un immense arbre de décision est construit de manière à fournir en ses feuilles, de nombreux sous-ensembles d'unités du niveau « diphone », pouvant appartenir à plusieurs corpus. Chacune de ces feuilles correspond à des sous-ensembles d'unités acoustiquement homogènes. À l'étape de synthèse, ces sous-ensembles sont accessibles *via* une succession de requêtes construites à partir du texte à synthétiser. Il en résulte des sous-ensembles d'unités candidates. On sélectionne parmi ces sous-ensembles, les unités qui minimisent une distance de concaténation, grâce à la programmation dynamique (Viterbi, 1967). Cette distance est, elle aussi, apprise automatiquement, et permet de favoriser le naturel des transitions au niveau segmental, mais aussi au niveau suprasegmental. Comme le langage de requête d'IrcamCorpusTools permet de stipuler des contraintes acoustiques, celles-ci peuvent être définies et ajoutées, de manière à influencer la prosodie finale de la phrase de synthèse. Ces contraintes prosodiques peuvent, par exemple, être fournies par un modèle de la proéminence ou par un modèle de l'expressivité. Un autre degré de liberté est fourni à l'utilisateur qui peut bannir certaines unités, afin que l'algorithme de sélection en choisisse d'autres parmi les sous-ensembles candidats possibles.

6. Conclusion

Dans cet article, nous avons présenté IrcamCorpusTools, une plate-forme extensible pour la création, la gestion et l'exploitation des corpus de parole. Elle permet facilement d'interfacer des données hétérogènes avec des analyseurs internes ou externes, en utilisant le principe d'autodescription des données et des analyseurs. En outre, l'autodescription des données garantit leur pérennité, favorise l'introduction de nouveaux types et leur confère une plus grande visibilité. De même, l'autodescription des analyseurs assure l'extensibilité de la plate-forme ainsi que sa modularité et la mutualisation des corpus. La plate-forme IrcamCorpusTools est capable de gérer les relations hiérarchiques multiples et séquentielles entre des unités. Un langage de requête simple et expressif donne un accès immédiat aux données de ces unités. Ces fonctionnalités appliquées à différents corpus de parole (parole contrôlée, parole spontanée pour des études de la prosodie et/ou de l'expressivité) intéressent directement les recherches à la frontière entre le traitement automatique des langues et le traitement automatique de la parole. En guise d'exemple, un processus de synthèse de parole ex-

pressive à partir du texte est décomposé en opérations élémentaires, depuis sa genèse, jusqu'au résultat sonore. L'intégration de ces exploitations énumérées au sein d'une même plate-forme, illustre les avantages de l'interopérabilité. Ceci doit être interprété comme un encouragement au partage des outils entre les communautés TAL et TAP. C'est pourquoi nous avons le projet de distribuer publiquement IrcamCorpusTools à ces communautés de chercheurs.

Remerciements

Les auteurs remercient Diemo Schwarz pour avoir posé certaines bases d'IrcamCorpusTools. Le développement d'IrcamCorpusTools est partiellement supporté par :

– le projet RIAM VIVOS[18] sur la création de voix expressives pour des applications multimédias ;

– le projet ANR Rhapsodie 07 Corp-030-01 [19] sur l'élaboration de corpus prosodiques de référence en français parlé (36 heures).

Bibliographie

Avanzi M., Lacheret-Dujour A., Victorri B., « ANALOR. A Tool for Semi-Automatic Annotation of French Prosodic Structure », *Proceedings of Speech Prosody 2008,* p. 119, 2008.

Barras C., Geoffrois E., Wu Z., Liberman M., « Transcriber : a Free Tool for Segmenting, Labeling and Transcribing Speech », *LREC*, p. 1373-1376, 1998.

Beller G., « Influence de l'expressivité sur le degré d'articulation », *RJCP, Rencontres Jeunes Chercheurs de la Parole*, p. 24-27, 2007a.

Beller G., « Transformation de la parole dépendante de l'expressivité et du texte », *Journée des Sciences de la Parole*, p. 45, 2007b.

Beller G., Marty A., « Talkapillar : outil d'analyse de corpus oraux », *Rencontres Jeunes Chercheurs de L'Ecole Doctorale 268*, Paris 3 Sorbonne-Nouvelle, p. 97-100, 2006.

Beller G., Obin N., Rodet X., « Articulation Degree as a Prosodic Dimension of Expressive Speech », *Speech Prosody 2008*, Campinas, p. 681-684, 2008.

Beller G., Rodet X., « Content-based transformation of the expressivity in speech », *Proceedings of the 16th ICPhS*, Saarbruecken, p. 2157-2160, August, 2007.

Beller G., Schwarz D., Hueber T., Rodet X., « Speech Rates in French Expressive Speech », *Speech Prosody 2006*, SproSig, ISCA, Dresden, p. 672-675, 2006.

Bilhaut F., Widlöcher A., « LinguaStream : An Integrated Environment for Computational Linguistics Experimentation », *11th Conference of the European Chapter of the*

18. VIVOS : `http://www.vivos.fr`
19. RHAPSODIE : `http://rhapsodie.risc.cnrs.fr`

Association of Computational Linguistics (Companion Volume), Trento, Italy, p. 95-98, 2006.

Bird S., Day D., Garofolo J., Henderson J., Laprun C., Liberman M., « ATLAS : A Flexible and Extensible Architecture for Linguistic Annotation », *in Proceedings of the Second International Conference on Language Resources and Evaluation*, p. 1699-1706, 2000.

Bird S., Liberman M., « A formal framework for linguistic annotation », *Speech Commun.*, vol. 33, n° 1-2, p. 23-60, 2001.

Boersma P., Weenink D., « Praat, a system for doing phonetics by computer », *Glot international*, vol. 5-9 of *10*, p. 341-345, 2001.

Bogaards N., Roebel A., Rodet X., « Sound Analysis and Processing with AudioSculpt 2 », *International Computer Music Conference (ICMC)*, Miami, USA, Novembre, 2004.

Cassidy S., Harrington J., « Multi-level annotation in the Emu speech database management system », *Speech Communication*, vol. 33, 1-2, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, p. 61-77, 2001.

Chafe W., « The importance of corpus linguistics to understanding the nature of language », *in* J. Svartvik (ed.), *Directions in Corpus Linguistics*, Proceedings of Nobel Symposium 82, Berlin - New York : Mouton de Gruyter, p. 79-97, 1992.

Cunningham H., Maynard D., Bontcheva K., Tablan V., « GATE : A framework and graphical development environment for robust NLP tools and applications », *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, p. 168-175, 2002.

d'Alessandro C., Mertens P., « Automatic pitch contour stylization using a model of tonal perception », *Computer Speech and Language*, vol. 9, p. 257-288, 1995.

De Cheveigné A., Kawahara H., « YIN, a Fundamental Frequency Estimator for Speech and Music », *JASA*, vol. 111, p. 1917-1930, 2002.

Degottex G., Bianco E., Rodet X., « Measure of glottal area on high-speed videoendoscopy », *Speech Production Workshop : Instrumentation-based approach*, p. 25-28, 2008a.

Degottex G., Bianco E., Rodet X., « Usual to particular phonatory situations studied with high-speed videoendoscopy », *The 6th International Conference on Voice Physiology and Biomechanics*, p. 19-26, 2008b.

Durand J., Laks B., Lyche C., « Un corpus numérisé pour la phonologie du français », *In G. Williams (ed.) La linguistique de corpus. Rennes : Presses Universitaires de Rennes*, p. 205-217, 2005.

Gussenhoven C., Jacobs H., *Understanding phonology*, Arnold, 2005.

Gut U., Milde J.-T., Voormann H., Heid U., « Querying Annotated Speech Corpora », *Proceedings of Speech Prosody 2004*, Nara, Japan, p. 569-572, 2004.

Habert B., « Des corpus repésentatifs : de quoi, pour quoi, comment ? », *Linguistique sur corpus*, Perpignan, France, p. 11-58, 2000.

Henrich N., Etude de la source glottique en voix parlée et chantée, PhD thesis, Université Paris 6, Paris, France, nov, 2001.

Hirst D., Cristo A. D., Espesser R., *Prosody : Theory and Experiment*, Kluwer Academic, M. Horne (ed), chapter Levels of representation and levels of analysis for intonation, p. 51-87, 2000.

Hirst D., Espesser R., « Automatic modelling of fundamental frequency using a quadratic spline function », *Travaux de l'Institut de Phonétique d'Aix*, vol. 15, p. 71-85, 1993.

Hunt A. J., Black A. W., « Unit selection in a concatenative speech synthesis system using a large speech database », *ICASSP*, IEEE Computer Society, Washington, DC, USA, p. 373-376, 1996.

Lai C., Bird S., « Querying and updating treebanks : A critical survey and requirements analysis », *In Proceedings of the Australasian Language Technology Workshop*, p. 139-146, 2004.

Lamel L., Gauvain J.-L., Eskénazi. M., « Bref, a large vocabulary spoken corpus for French », *EuroSpeech*, p. 505-508, 1991.

Lanchantin P., Morris A. C., Rodet X., Veaux C., « Automatic Phoneme Segmentation with Relaxed Textual Constraints », *in* E. L. R. A. (ELRA) (ed.), *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008.

MacWhinney B., *The CHILDES project : Tools for analyzing talk, Third edition*, vol. Volume I : Transcription format and programs, Lawrence Erlbaum Associates, Mahwah, NJ, 2000.

Müller C., « A flexible stand-off data model with query language for multi-level annotation », *Annual Meeting of the Association for Computational Linguistics*, p. 109-112, 2005.

Nakov P., Schwartz A., Wolf B., Hearst M., « Supporting annotation layers for natural language processing », *Annual Meeting of the Association for Computational Linguistics*, p. 65-68, 2005.

Obin N., Goldman J., Avanzi M., Lacheret-Dujour A., « Comparaison de 3 outils de détection automatique de proéminence en français parlé », *XXVIIème Journées d'Études de la Parole*, Avignon, France, p. 153-157, 2008a.

Obin N., Lacheret-Dujour A., Veaux C., Rodet X., Simon A.-C., « A Method for Automatic and Dynamic Estimation of Discourse Genre Typology with Prosodic Features », *Interspeech 2008*, Brisbane, Australia, 2008b.

Obin N., Rodet X., Lacheret-Dujour A., « French Prominence : a Probabilistic Framework », *proc. of ICASSP*, Las Vegas, Nevada, USA, p. 3993-3996, 2008c.

Oostdijk N., « The Spoken Dutch Corpus : Overview and first evaluation », *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'2002)*, p. 887-893, 2000.

Pfitzinger H., « Five Dimensions of Prosody : Intensity, Intonation, Timing, Voice Quality, and Degree of Reduction », *in* H. Hoffmann, R. ; Mixdorff (ed.), *Speech Prosody*, n° 40 in *Abstract Book*, Dresden, p. 6-9, 2006.

Rabiner L., « A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition », *IEEE*, vol. 77, 2, p. 257-286, 1989.

Rosenberg A., Hirschberg J., « Detecting Pitch Accent Using Pitch-corrected Energy-based Predictors », *Proceedings of Interspeech 2007*, Antwerp, Belgium, p. 2777-2780, 2007.

Sjölander K., Beskow J., « WaveSurfer - An Open Source Speech Tool », *International Conference on Spoken Language Processing*, vol. 4, Beijing, China, p. 464-467, 2000.

Taylor P., Black A. W., Caley R., « Heterogeneous Relation Graphs as a Mechanism for Representing Linguistic Information », *Speech Communication*, vol. 3, p. 153-174, January, 2001.

Veaux C., Beller G., Rodet X., « IrcamCorpusTools : an Extensible Platform for Spoken Corpora Exploitation », *in* E. L. R. A. (ELRA) (ed.), *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may, 2008.

Vincent D., Rosec O., Chonavel T., « Estimation of LF glottal source parameters based on an ARX model », *9th European Conference on Speech Communication and Technology*, Lisbonne, Portugal, p. 333-336, 2005.

Viterbi A. J., « Error bounds for convolutional codes and an asymptotically optimum decoding algorithm », *IEEE TIT*, vol. 13(2), p. 260-269, 1967.