



OOPS: une approche orientée objet pour l'interrogation et l'analyse linguistique de l'interface prosodie/syntaxe/discours

Julie Beliao, Antoine Liutkus

► To cite this version:

Julie Beliao, Antoine Liutkus. OOPS: une approche orientée objet pour l'interrogation et l'analyse linguistique de l'interface prosodie/syntaxe/discours. 4e Congrès Mondial de Linguistique Française, Jul 2014, Berlin, Allemagne. pp.2565-2581, 10.1051/shsconf/20140801273 . hal-01053422

HAL Id: hal-01053422

<https://hal.archives-ouvertes.fr/hal-01053422>

Submitted on 30 Jul 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

OOPS: une approche orientée objet pour l'interrogation et l'analyse linguistique de l'interface prosodie/syntaxe/discours

Beliao, Julie & Liutkus, Antoine

Laboratoire MoDyCo, UMR 7114, Université Paris Ouest, France
Inria, PAROLE, France

julie@beliao.fr, antoine.liutkus@inria.fr

1. Introduction

Dans cet article, nous nous intéressons à la problématique de l'étude de la langue parlée multi-annotée. Dans de tels corpus, un même échantillon de parole est associé à des informations inhérentes à différents niveaux linguistiques. Cela soulève des problématiques liées à la difficulté d'organisation, de stockage et d'accès à ces informations pour l'analyse conjointe de niveaux linguistiques : intonosyntaxe, discours-prosodie et syntaxe-pragmatique par exemple. La principale difficulté qui sous-tend l'exploitation d'un tel corpus multi-annoté de langue parlée est la mise en relation d'unités qui appartiennent à des niveaux linguistiques différents. Pour tous les niveaux linguistiques représentés, chaque annotation conduit en effet à une hiérarchie particulière.

L'agrégation de toutes ces hiérarchies ou arborescences linguistiques est l'enjeu du formalisme proposé. Pour étudier l'interface entre différents niveaux linguistiques, nous proposons une approche orientée objet OOPS (*Object-Oriented Processing of Speech*) permettant de représenter une large variété d'annotations au sein d'une architecture globale. Une telle structure ne peut en effet pas être réalisée entièrement à partir de la seule transcription annotée comme celle présentée en Table 1, qui est au mieux exploitable par un humain. Elle nécessite au contraire une mise en relation du signal et des autres support d'annotation avec cette transcription pour l'étude conjointe d'unités linguistiques appartenant à des niveaux différents. La particularité de l'approche que nous proposons est qu'elle repose entièrement sur un formalisme *modulaire*, ou *objet*. Une unité linguistique sera vue comme un objet (au sens informatique du terme) de la hiérarchie dépendant du niveau linguistique auquel elle appartient. Ces différentes hiérarchies sont reliées par les mots de la transcription, qui leurs sont communs. Ainsi, il devient possible de faire des requêtes mettant en jeu plusieurs niveaux linguistiques : syntaxe-prosodie, syntaxe-pragmatique ou encore prosodie-pragmatique, pour en extraire toute information jugée pertinente.

L'approche que nous proposons repose sur le postulat que plus l'information sera modulaire, plus son traitement en sera simple et puissant. Cette hypothèse nous a conduit à envisager certaines structures sous un angle un peu différent de celui proposé par les membres du projet Rhapsodie dans un soucis de toujours plus modulariser l'information linguistique.

D'un point de vue beaucoup plus pratique, le système que nous décrivons dans cet article a été développé sous la forme d'un module Python permettant l'analyse et l'exploitation de données annotées selon le système mis en place dans le cadre du projet Rhapsodie (Lacheret, Kahane &

Pietrandrea (eds) à paraître). L'ensemble des outils présentés dans cette étude est diffusé sous licence libre GPLv3 et peut être téléchargé à l'adresse www.loria.fr/~aliutkus/oops

```
le mécanisme { tout simplement | pas spécialement de la carrière } //+ | ^et la personnalité
aussi } // j'aime bien comprendre comment ça marche > les autres // ça < c'est "euh" & //
"bon" je l'ai toujours eu "je crois" // par conséquent < j'ai eu envie de savoir un petit peu
mieux comment ça marchait // ^et ^puis je suis toujours étonnée //+ maintenant < plus // ^et je
précise d'ailleurs que [ quand Marcel Achard a écrit ce texte <+ il y avait déjà presque dix-sept
ans ] //
```

Table 1: *Exemple de texte annoté en macrosyntaxe, entassements et unités illocutoires extrait de [Rhap-D2001, Corpus Mertens (Mertens 1987)]*

Cet article est structuré comme suit. Tout d'abord, nous présentons en partie II les différentes structures linguistiques en jeu. En partie III, nous introduisons le formalisme adopté pour la représentation informatique de ces objets linguistiques. Ensuite, nous montrons en partie IV comment cette structure peut être exploitée pour réaliser différents traitements utiles à la linguistique de corpus, comme la stylisation prosodique d'unités pragmatiques ou des décomptes intonosyntaxiques.

2. Présentation des différentes structures prises en compte

Un certain nombre d'annotations différentes ont été réalisées dans le cadre du projet Rhapsodie¹. Les systèmes d'annotation considérés sont ceux qui ont été introduits par les participants du projet Rhapsodie, présentés dans (Avanzi et al 2011, Benzitoun et al. 2009, 2010, à paraître, Lacheret et al. 2011) et dont le corpus est disponible en libre-accès sur internet. La grande richesse et variété des annotations réalisées dans le cadre du projet Rhapsodie en fait le support idéal pour mettre en œuvre l'approche objet OOPS que nous décrivons ici. Nous avons choisi d'interpréter ces annotations selon trois perspectives linguistiques différentes : (i) le niveau syntaxique, (ii) le niveau pragmatiqueⁱⁱ et (iii) le niveau prosodique. Parmi les annotations de chacun de ces niveaux, on en recense trois types que nous présentons dans les sous-sections qui suivent. Pour le niveau syntaxique : *macrosyntaxe*, *entassements* (Benzitoun et al. 2009, 2010, Gerdes et Kahane 2010; et une vingtaine pour le niveau prosodique (Avanzi 2011) : *phonèmes*, *syllabes*, *mots*, *pièdes*, *groupes*, *paquets*, *périodes intonatives*, *proéminences*, *disfluences*, *tours de parole*, etc. Ces annotations ont ensuite donné lieu à une interprétation informatique qui nous est propre. Les structures informatiques implémentées que nous présentons donc dans la suite de cet article ne sont donc potentiellement pas identiques à celles présentées dans le cadre du projet Rhapsodie.

2.1. Le niveau syntaxique

L'annotation syntaxique des transcriptions du corpus Rhapsodie ont été effectuées à la main par des spécialistes en syntaxe. Cette annotation se présente sous la forme d'un balisage avec des symboles spécifiques rajoutés à la transcription (voir table 2) et permet l'encodage de tous les phénomènes considérés comme faisant partie de la syntaxe dans le cadre du projet, c'est-à-dire

la macrosyntaxe (Benzitoun et al. 2009, 2010) et les phénomènes d'entassements (Gerdes et Kahane 2010).

Le balisageⁱⁱⁱ présenté en table 1 encode plusieurs phénomènes syntaxiques en même temps. Ainsi selon l'interprétation que l'on fait de ce balisage, il est possible de générer plusieurs arborescences différentes. En figure 1 on peut voir une arborescence macrosyntaxique, les feuilles de l'arbre sont les mots de la transcription qui sont regroupés en fonction des constituants macrosyntaxiques auxquels ils appartiennent formant ce que l'on appelle des *bags of words* (i.e. des ensembles de mots). Cet arbre est le résultat d'une interprétation macrosyntaxique du balisage, macrosyntaxique c'est-à-dire en ignorant les symboles relatifs à d'autres arborescences syntaxiques. Chaque échantillon du corpus est analysé selon cet angle créant autant d'arbres qu'il y a d'unités macrosyntaxiques englobantes (EMU). Ces arbres possèdent ensuite autant de branches que l'EMU contient de composantes macrosyntaxiques (MC), chaque MC peut avoir un type différent dépendant de sa fonction macrosyntaxique. Enfin les enfants ou feuilles des MC sont les mots de la transcription, qui se retrouvent ainsi regroupés selon leur parenté. Parmi les phénomènes macrosyntaxiques représentés en figure 1, on compte trois MC: deux *noyaux* (//), un *post-noyau* (<) soit trois *bags of words* et deux EMU (//).

- (1) le mécanisme tout simplement pas spécialement de la carrière et la personnalité aussi // j'aime bien comprendre comment ça marche > les autres // [Rhap-D2001, Corpus Mertens]

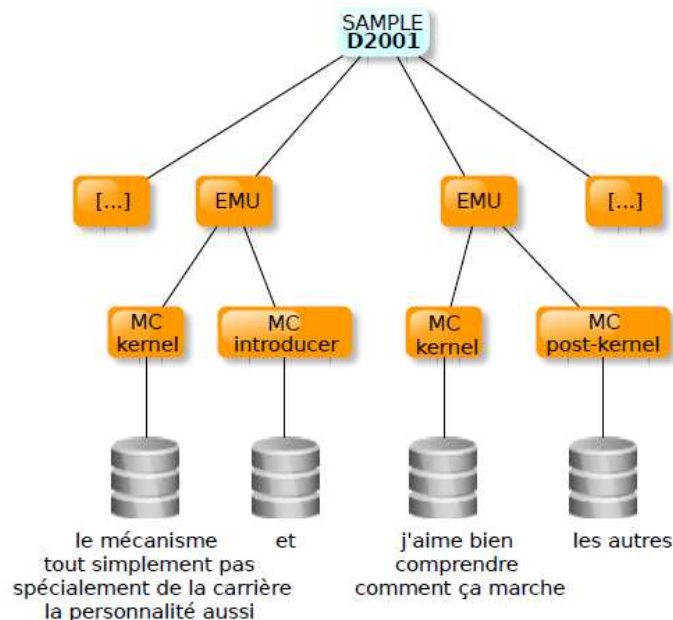


Figure 1: Représentation arborescente macrosyntaxique des unités macrosyntaxiques englobantes (EMU) et des composantes macrosyntaxiques (MC).

Nous avons détaillé ci-dessus l'interprétation du balisage selon une perspective macrosyntaxique, nous allons voir maintenant que l'on peut interpréter le balisage selon une

autre perspective. Si l'on choisit cette fois de construire des arbres d'entassement, il suffit alors d'ignorer tous les symboles du balisage qui n'encodent pas les phénomènes d'entassements (Blanche-Benveniste 1990, Bonvino et al. 2009, Gerdes et Kahane 2011, Kahane et Pietrandrea 2012). Les symboles d'entassements permettent de délimiter des constituants syntaxiques qui peuvent occuper la même position syntaxique régie, c'est-à-dire pouvant assumer les mêmes fonctions syntaxiques.

- (2) { le mécanisme { tout simplement | pas spécialement de la carrière } | ^et la personnalité aussi } j'aime bien comprendre comment ça marche les autres [Rhap-D2001, Corpus Mertens]

Si nous reprenons la même portions de transcription annotée que pour l'exemple (1) en ne conservant cette fois que les symboles marquant le phénomène d'entassement nous pouvons constater la présence d'un entassements à deux couches dont la première couche contient un autre entassement à deux couches. Nous distinguons donc deux types de membres, i.e. l'entassement en lui-même, qui déclenche la création d'un arbre d'entassement et les couches d'entassement qui sont alors les enfants de ces derniers. Ainsi on observe que le premier entassement est composé de deux couches, la couche « *le mécanisme tout simplement pas spécialement de la carrière* » qui peut être substituée par la couche « *la personnalité aussi* » et on distingue également à l'intérieur de cette première couche un deuxième entassement lui-même composé de deux couches, la « *tout simplement* » qui peut être substituée par la couche « *pas spécialement de la carrière* ». De ce fait et de la même manière que pour l'arbre de la macrosyntaxe, nous obtenons des arbres d'entassement dont les feuilles sont aussi les mots de la transcription formant cette fois-ci des *bags of words* spécifiques à la perspective d'entassement (voir figure 2).

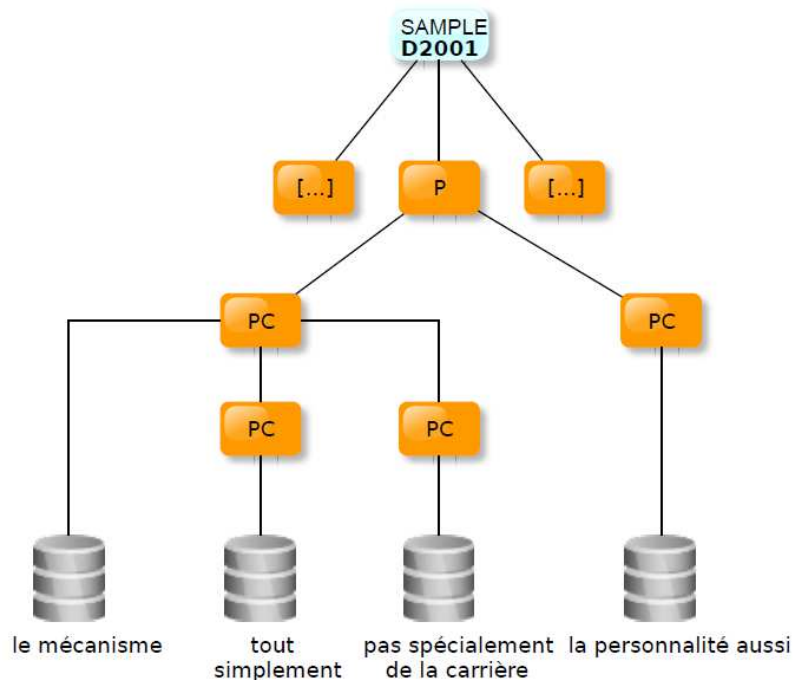


Figure 2: Représentation arborescente des entassements (P) et des couches d'entassement (PC).

2.2. Le niveau pragmatique

Nous avons présenté dans la section précédente de quelle manière nous interprétons le balisage réalisé manuellement pour construire deux types d'arbres syntaxiques différents, les arbres macrosyntaxiques et les arbres d'entassement. Ces deux types d'arbres ou hiérarchies sont ceux qui ont été également pris en compte dans le cadre du projet Rhapsodie. Cela-dit l'introduction d'une nouvelle hiérarchie nous a paru intéressante dans la mesure où cela contribue à modulariser encore plus l'architecture générale des données du corpus facilitant à postériori le requêtage sur ces données. Cela nous a donc amené à considérer le balisage manuel selon une perspective supplémentaire qui se trouve être une perspective rendant compte de niveau pragmatique.

- (3) le mécanisme tout simplement pas spécialement de la carrière //+ et la personnalité aussi // j'aime bien comprendre comment ça marche les autres // [Rhap-D2001, Corpus Mertens]

Le balisage manuel de la transcription compte également le marquage des *unités illocutoires*, ces éléments se ramènent à ce que la littérature qualifie traditionnellement d'acte de langage ou acte illocutoire (Searl 1969 et 1975, Austin 1962). Dans le cadre du projet, ces unités sont considérées comme faisant partie de la macrosyntaxe et sont donc incluses dans les arbres macrosyntaxiques se confondant alors avec les segments que nous avons appelé *unités macrosyntaxiques englobantes* (EMU). Ceci nous amène donc à introduire un troisième type

d'arbre, les arbres *illocutoires* qui sont alors relatifs à la pragmatique. L'appartenance de ces arbres au niveau pragmatique se justifie par la nature des segments qui les composent. Effectivement, chaque échantillon du corpus est analysé en ne tenant compte que des symboles marquant les *unités illocutoires* créant autant d'arbres illocutoires. Ces arbres extrêmement simples ont ensuite pour enfant les mots de la transcription qui leur correspondent, formant de cette façon un nouveau type de *bags of words* propre à ce niveau. Nous construisons donc un arbre *illocutoire* par échantillon, dont les enfants, les unités illocutoires, sont détectés par les symboles // et //+^{IV}. Si nous prenons l'exemple (3) on compte alors trois unités illocutoires distinctes (voir figure 3).

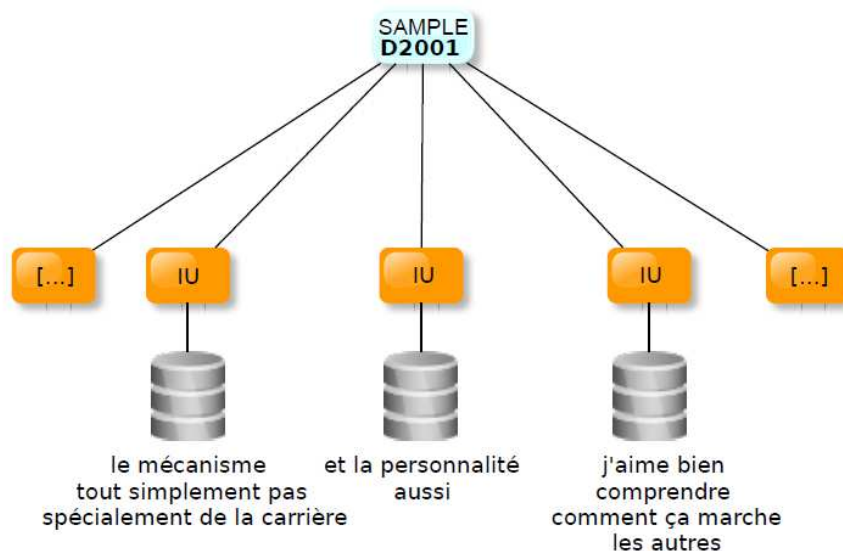


Figure 3: Représentation arborescente des unités illocutoires (IU).

2.3. Le niveau prosodique

Nous avons vu ci-dessus que la transcription du signal de parole a fait l'objet d'un balisage manuel qui a permis la construction des arborescences des niveaux syntaxique et pragmatique. Cela dit, la transcription annotée n'est pas le seul support qui encode de l'information linguistique. Les annotations d'ordre prosodique par exemple ont été réalisées dans un TextGrid qui est le format de fichier du logiciel Praat, logiciel qui a permis d'encoder l'information prosodique. On dispose ainsi d'une segmentation en syllabes et en phonèmes alignée au signal de parole (Goldman 2011) ainsi que d'un certain nombre d'annotations (Avanzi 2010) : proéminences, disfluences ; et segmentations en unités prosodiques : périodes intonatives (Lacheret et Victorri 2002), paquets intonatifs, groupes rythmiques et pieds. Nous avons à partir de ces informations dérivé un quatrième type d'arbre (figure 4), les arbres prosodiques, dont les mots sont de nouveau les feuilles et forment des *bags of words* propre à l'arborescence prosodique.

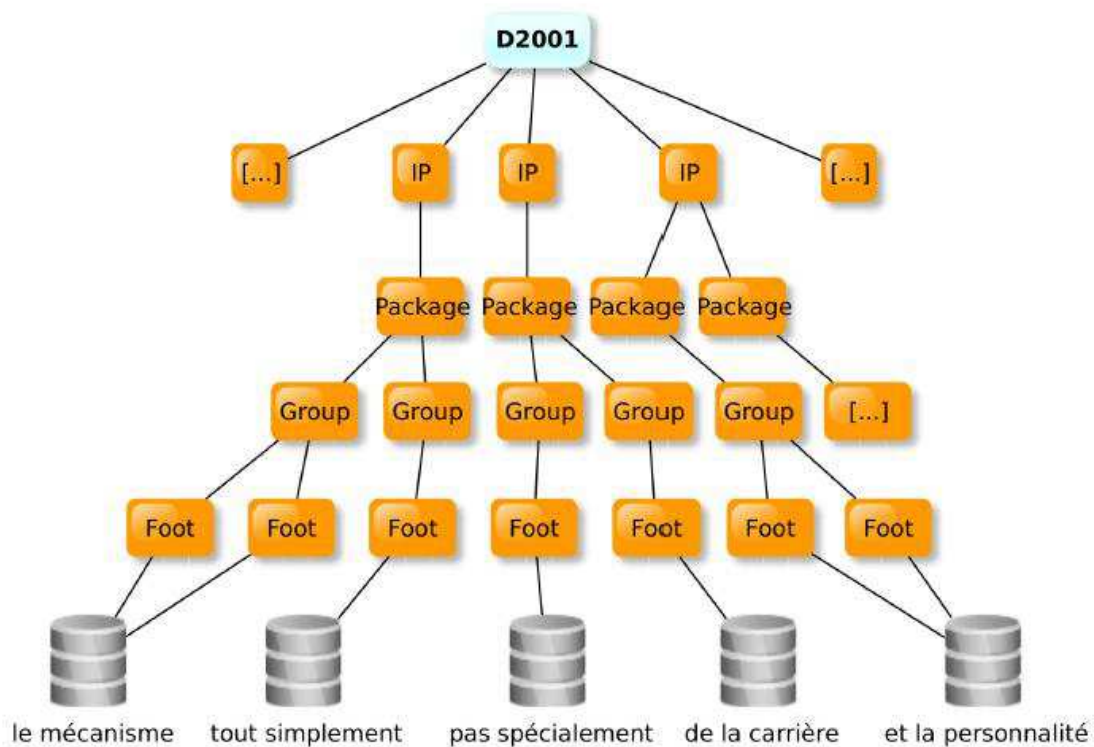


Figure 4: Représentation arborescente des unités prosodiques, périodes intonatives (IP), paquets intonatifs (Package), groupes rythmiques (Group), pieds (Foot).

Nous avons vu que pour ces quatre arborescences l'unité linguistique minimale ou feuille est le mot de la transcription. Cela-dit, il existe des unités inférieures au mot, comme par exemple la syllabe ou le phonème, ces unités ne sont pas oubliées et sont intégrées à notre structure de donnée en tant qu'information ou étiquette typant les mots. Ce choix nous est apparu comme étant le meilleur compromis car il permet notamment qu'une même étiquette (par exemple une syllabe) viennent typer deux mots (permettant ainsi de rendre compte des phénomènes de liaison). Ajoutons aussi que strictement chaque unité linguistique intégrée à notre architecture peut-être typée en fonction des ces caractéristiques temporelles, positionnelles (ce qui permet entre autre de ne pas perdre la séquentialité des unités) et acoustique (grâce à l'alignement transcription/signal).

3. Formalisme adopté : l'approche OOPS

3.1. Une approche objet

Chaque arborescence linguistique présentée dans la section précédente fait intervenir des unités qui lui sont propres et qui sont liées par des relations structurelles. Dans le formalisme objet que nous proposons, chacune sera modélisée comme un *objet*. Ainsi, un entassement sera un objet

auquel sont reliées ses différentes couches. Une unité englobante macrosyntaxique est également un objet, relié à différents objets rendant compte d'unités telles que les noyaux, post-noyaux, etc. Enfin, d'un point de vue prosodique, le signal de parole est segmenté en intervalles de taille variable qui peuvent être inclus les uns dans les autres. Dans tous les cas, le point commun ou point d'ancrage de ces différents objets se fait par les mots de la transcription.

Les mots ou tokens (représentés par une zone rouge sur la figure) de la transcription sont ainsi connectés à un ensemble d'objets linguistiques, chacun appartenant à une ou plusieurs arborescences de niveaux linguistiques particulier (les quatre hiérarchies sont représentées par des exemples d'arbres miniaturisés présentés en partie II). Ainsi comme on le voit en figure 5, la structure obtenue représente pour un même ensemble de mots de la transcription autant d'arbres différents que de niveaux ou perspectives linguistiques considérées. Une telle architecture ne peut pas se réduire à une unique structure arborée. Au contraire, elle permet d'associer directement plusieurs arbres linguistiques, tous reliés entre eux par le biais des mots de la transcription, ou plus exactement des tokens.

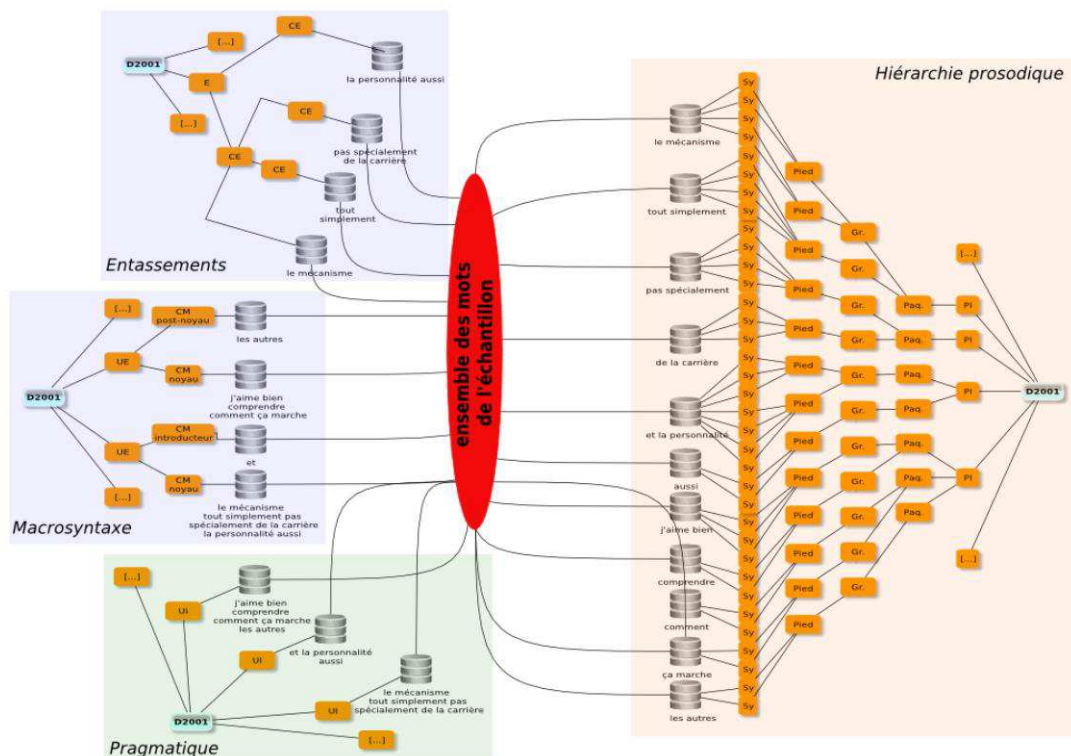


Figure 5: Mise en relation des différentes arborescences linguistiques considérées : prosodique, macrosyntaxique, entassement et pragmatique par l'intermédiaire des tokens (dans l'amas rouge)

D'une manière générale, nous considérons que les unités linguistiques que nous étudions peuvent être représentés par des objets au sens informatique du terme. Chacun de ces objets est alors lié avec les autres. Par exemple, un entassement ayant deux couches correspondra à un objet nœud («Node») de type *entassement*, auquel sont attachés deux objets nœuds de type

composante d'entassement. Chaque mot de la transcription sera donc représenté par un objet « Token », qui a pour attribut sa position, son libellé, etc. Si un objet syntaxique contient deux mots de la transcription, il sera relié à chacun des objets « Token » correspondants. Si une unité prosodique contient ces mêmes mots, ce sera vers les mêmes objets qu'elle sera reliée, etc.

La force de ce *formalisme objet* est que l'information n'est pas dupliquée mais qu'au contraire les dépendances sont encodées directement sous la forme de liens entre structures de données. Cette approche permet de directement encoder l'information linguistique telle qu'elle est formalisée ici, sans la nécessité d'utiliser un système d'encodage intermédiaire comme le XML, qui est surtout adéquat pour le stockage d'arborescences simples. Le rajout de nouveaux objets linguistiques à l'architecture générale peut dès lors se faire facilement en rajoutant des liens vers les objets déjà existants, sans se préoccuper dans autres dépendances déjà encodées ou de modifier le format d'encodage, ou de dupliquer l'information déjà existante. Cette propriété permet un développement incrémental et modulaire de l'architecture qui assure la pérennité des informations déjà encodées. C'est ce qu'on appelle la propriété d'*encapsulation* de l'approche objet.

3.2. Les différentes classes considérées

En informatique orientée objet, chaque objet est dit instance d'une *classe*. Cette classe définit les *propriétés communes* des objets qui en sont l'instance, ici des unités linguistiques. Elle déclare des *attributs* spécifiques de même que des *méthodes*, qui sont autant de comportement qui varient d'une classe d'objet à l'autre. Une classe peut aussi être vue comme un *moule* ou une *usine* à partir de laquelle il est possible de créer de nouveaux objets. Dans notre cas, nous définirons par exemple une classe nœud, dont les différentes unités linguistiques seront des instances. Cette classe présente comme attribut le type d'unité (pile, unité illocutoire, etc), ainsi que les autres unités auxquelles elle est reliée dans son arborescence. Plusieurs arborescences seront ainsi définies, une pour chaque niveau linguistique considéré. Cette capacité de généralisation permet de prendre en charge un grand nombre d'arborescences tout à fait différentes sans avoir à changer l'ensemble du système mis en place. Par ailleurs, les objets peuvent être manipulés de façon simple car, bien que structurellement différent, les opérations de requête les concernant sont similaires.

3.2.1. Pour représenter des mots de la transcription: la classe d'objets « Token »

Le texte annoté se présente sous la forme de mots ou *tokens*. Un mot est défini comme un ensemble de caractères, séparés des autres par un espacement. En jargon informatique, nous dirons qu'un mot de la transcription annotée est un *objet instance de la classe* « Token » (Rumbaugh 1990, Booch et al. 1996). Chaque objet de cette classe aura en particulier un attribut *libellé*, ainsi qu'un attribut *position*.

3.2.2. Pour des unités syntaxiques ou pragmatiques : la classe d'objets Node

Chaque niveau d'analyse distinct conduit alors à la définition d'unités spécifiques. Toutes les unités syntaxiques et pragmatiques par exemple peuvent être comprises comme des objets instance de la même classe *nœud* (« Node ») qui ont tous en commun le fait d'avoir un *parent* (« parent »), des enfants (« childs »), un *type* particulier (pile ou unité illocutoire par exemple) et enfin un *contenu* (« content »), qui est une liste de mots auxquels ils sont reliés. L'utilisation combinée des classes Node et Token permet ainsi d'encoder conjointement l'ensemble des arborescences linguistiques que nous avons décrites en partie II. Il suffit pour ce faire de créer autant d'objets que nécessaire et d'établir entre eux les liens désirés. Cela est réalisé automatiquement à partir de la transcription annotée grâce à un module parseur ad-hoc implémenté. Sur la figure 1, nous voyons ainsi les objets Token (en gris) et les objets Node (en orange) former les hiérarchies d'entassement, de macro-syntaxe, de pragmatique.

3.2.3. Pour des unités prosodiques : la classe d'objets TextGrid, Tier et Interval

La seule considération des mots (Token) et des nœuds (Node) suffit à représenter la transcription annotée d'une manière exploitable pour l'analyse. Cependant, cette transcription annotée n'est pas seule : elle est produite à partir d'un enregistrement audio. Des analyses prosodiques peuvent être opérées sur le signal de parole, ce qui revient à le découper en *intervalles* (« Interval ») successifs. Chacun de ces découpages de l'échantillon porte le nom de Tier. De manière à rendre compte fidèlement de cet usage dans le formalisme proposé, chaque échantillon est associé à un objet de classe TextGrid, et qui contient plusieurs objets Tier. Chacune de ces Tier est une liste d'objets Interval, dont chacun a pour attributs sa *position initiale* (« xmin »), sa *position finale* (« xmax ») et son *libellé* (« mark »). Lorsqu'on considère une transcription annotée, il est clair que chaque mot prononcé correspond à une portion précise du signal audio considéré. Grâce à des outils tels qu'EasyAlign (Goldman 2012), il est possible d'identifier quelles portions du signal de parole correspondent à chaque mot de la transcription. Ainsi, l'introduction des classes TextGrid, Tier et Interval dans le formalisme permet d'inclure dans la représentation toutes les analyses effectuées par des outils manipulant des fichiers TextGrid provenant de Praat (Boersma et Weenink 2012). Cette procédure d'appariement est réalisée automatiquement grâce à notre implémentation de OOPS, et nécessite une correspondance parfaite entre le texte annoté de la transcription et la segmentation en tokens alignés signal dans le TextGrid réalisé par EasyAlign.

3.3. Utilisation

À partir de la transcription annotée et d'un fichier TextGrid qui contient le texte aligné par EasyAlign ainsi que toutes les analyses prosodiques désirées, le formalisme OOPS construit automatiquement une structure complète. Il est important de souligner ici que ces objets qui ont des attributs spécifiques ainsi que des liens de dépendance peuvent être stockés et interrogés *tels quels* par un langage orienté objet (Python ici). Autrement dit, il est possible aujourd'hui de stocker et de constituer des bases de données objet en lieu et place de bases de données

relationnelles classiques de type SQL. Il n'est pas nécessaire de repenser tout le formalisme de stockage si on souhaite rajouter certains objets à l'enregistrement.

Outre la facilité avec laquelle le formalisme objet permet la sauvegarde pérenne d'architectures linguistiques complexes, sa principale force réside surtout dans la souplesse des requêtes qu'il permet. Pour un échantillon donné, l'utilisateur a accès aux résultats de l'analyse par le biais de la structure dont nous présentons une illustration en figure 6. Il est alors très facile de rechercher de quelle manière des objets appartenant à différents types d'arborescences interagissent, permettant ainsi des études de l'interface syntaxe-prosodie, prosodie-discours, etc. Pour le moment, aucune interface graphique n'étant implémenté, ces requêtes prennent la forme de scripts en Python, dont nous fournissons de nombreux modèles pré-programmés de requêtes accessibles en ligne. Dans la suite de cet exposé, nous montrons comment l'approche OOPS a été mise en œuvre dans le cadre de deux études mettant en jeu des niveaux linguistique différents.

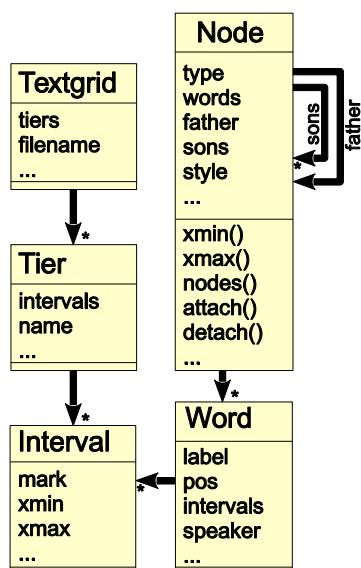


Figure 6: Diagramme de classe simplifié avec quelques-unes des principales classes de la structure obtenue avec l'approche OOPS. L'étoile (*) indique un nombre potentiel de lien entre les objets allant de zéro à plusieurs.

4. Présentation d'une requête d'interface

Comme nous l'avons présenté dans les sections précédentes, le formalisme proposé permet une manipulation efficace des unités linguistiques encodées dans le corpus. Nous proposons ici, deux cas d'étude mettant en avant la possibilité de réaliser une analyse d'interface pragmatique-prosodie et syntaxe-prosodie. Le premier consiste à récupérer la stylisation de la F0 pour chacune des unités illocutoires du niveau pragmatique. Le second cas d'interface nous permet d'observer si disfluença prosodique et disfluença syntaxique sont systématiquement corrélées.

4.1. Intégration de SLAM : un algorithme de stylisation de la F0

Dans le cadre du projet Rhapsodie, la méthode SLAM (*Stylization and Labelling of speech Melody*) a été développée dans le but de réaliser une stylisation automatique des contours de F0 de diverses unités linguistiques (Obin et al. soumis). L'algorithme de la méthode utilisée ici présente certains avantages par rapport aux autres méthodes existantes (Bailly et Aubergé 1995, Campione et al. 2000, Syrdal et al. 2001 ou encore Mertens 2004). D'une part, le système mélodique (c'est à dire, l'alphabet des contours mélodiques) est automatiquement calculé à partir du signal de parole. D'autre part, une simple représentation temps-fréquence est utilisé pour décrire des contours mélodiques complexes avec un alphabet limité de contours mélodiques^v et enfin la représentation proposée n'est pas limitée à un segment particulier de la parole mais peut couvrir n'importe quelle unité linguistique.

L'algorithme SLAM a été intégré à notre plateforme et le calcul de la stylisation peut se faire sur n'importe quel objet de la base, pour peu qu'il soit possible de remonter de cet objet à un support temporel et donc à une information de pitch et de locuteur. Le formalisme objet présenté en partie III nous a ainsi permis d'appliquer SLAM à l'ensemble des objets linguistiques et de leur attacher une stylisation de leur contour mélodique.

4.2. Utilisation de SLAM sur les unités stockées avec OOPS

L'objectif de cette démonstration de l'utilité de l'approche OOPS pour l'étude d'interfaces linguistiques est de présenter les vingt contours les plus courants dans la stylisation de la F0 des unités illocutoires. Pour cela, il est nécessaire d'isoler l'ensemble des objets *unités illocutoires*, la valeur de leurs bornes temporelles et de la F0 qui les caractérisent. Cette opération qui peut de prime abord sembler fastidieuse étant donné que les unités illocutoires ont pour support la transcription et non pas le signal, a été rendue possible et immédiate par le parcours des graphes de la plateforme, nous permettant ainsi de naviguer du niveau pragmatique au niveau prosodique.

On constate que sur les 400 contours possiblement générés par SLAM, 20 seulement suffisent pour caractériser plus de 80% des unités illocutoires (figure 9). C'est un résultat tout fait encourageant pour l'algorithme de stylisation d'une part et d'autre part cela attire notre attention sur les contours en question, qu'il semble intéressant d'observer de plus près pour les comparer à ceux suggérés par les études théoriques de l'intonation. Le premier modèle de représentation de l'intonation du français en termes de contours globaux (Delattre 1966) a porté sur l'intonation modale: un contour mélodique global spécifiant les modalités d'une phrase (soit les unités illocutoires ici). Plus récemment, ce paradigme a été étendu à la représentation de l'intonation comme la superposition de contours mélodiques élémentaires sur différentes unités linguistiques (Aubergé 1991).

Les résultats tendent à confirmer l'hypothèse faite par Delattre dans son article de 1966. Delattre y décrit les dix intonations de base de français. Dans le cadre qu'il a posé, ces intonations sont commutables entre elles et discrètes car elles génèrent des changements de sens. Or, si on observe les résultats obtenus en figure 9, notamment les dix premiers contours du classement (voir Table 3), on remarque qu'ils permettent de styliser à eux-seuls, la F0 d'environ 70 % des unités illocutoires.

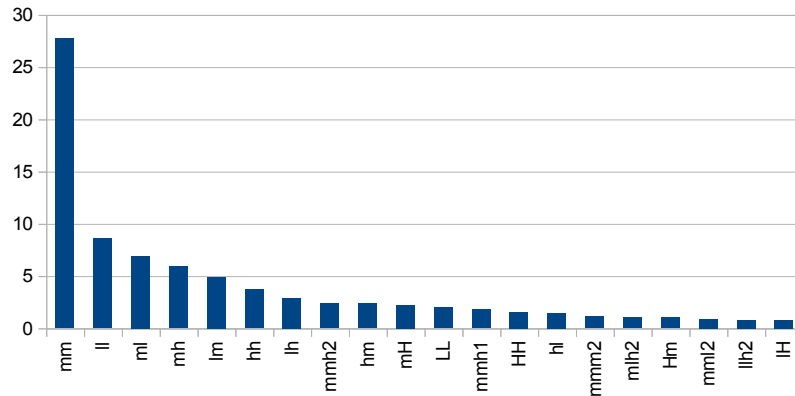


Figure 7: Les 20 contours les plus courants pour décrire les unités illocutoires (en pourcentages)

mm (28%)		Je fais quelques courses [Rhap-D0001, CFPP2000 (Branca et al. 2012)]
ll (9%)	<i>parenthèse</i>	C'est cet Aligre-là [Rhap-D0001, CFPP2000]
ml (7%)	<i>finalité</i>	Donc en fait euh voilà [Rhap-M1001, Professionnel]
mh (6%)	<i>continuation mineure</i>	Bah ça paraît normal [Rhap-D0002, CFPP2000]
lm (5%)	<i>commandement, exclamation</i>	Alors tout de même Gallas a pris un carton je vous signale [Rhap-D2003, PFC]
hh (4%)	<i>écho</i>	Alors Magalie est avec nous [Rhap-D2011, Broadcast]
lh (3%)	<i>continuation majeure</i>	C'était lui oui [Rhap-D2003, PFC]
mmh2 (2,5%)	<i>implication</i>	Ça sera un coup-franc Eugène [Rhap-D2003, PFC]
hm (2,5%)	<i>interrogation</i>	Alors pour le refaire tu me dis qu'il faudrait que j'aille vers Isigny [Rhap-D0009, PFC]
mH (2%)	<i>question</i>	Alors il me referaient l'assise [Rhap-D0009, PFC]

Table 1: Seulement 10 contours stylisés permettent de représenter environ 70 % des unités illocutoires du corpus utilisé. En italique sont indiqués le nom des intonations de base chez Delattre. (L) très bas, (l) bas, (m) medium, (h) haut, (H) très haut, (1,2,3) indiquent la position de la saillance.

La lecture de la table 3 nous apprend que les stylisations de F0 des unités illocutoires du corpus Rhapsodie correspondent de manière assez satisfaisante aux descriptions de Delattre. Ceci n'est cependant qu'une étude préliminaire et une analyse plus approfondie comme celle menée par [Schriberg 1998] permettrait peut-être un typage automatique des unités illocutoires sur lesquelles nous travaillons.

Un certain nombre d'autres études d'interfaces utilisant la plateforme ont été menées jusqu'à présent, on peut citer par exemple une étude d'interface intonosyntaxique axée sur l'analyse conjointes des disfluences syntaxiques et prosodique (Beliao 2013a) ou encore une étude axée sur la caractérisation du genre textuel d'échantillons de français parlé et basée sur l'observation de frontières pragmatiques et prosodiques (Beliao 2013b).

5. Conclusion

Dans cet article, nous avons montré que l'étude de la langue parlée implique de pouvoir manipuler simultanément des informations appartenant à des niveaux linguistiques différents, comme la syntaxe, la prosodie ou la pragmatique. Nous avons aussi évoqué le besoin de plus en plus important pour les chercheurs de se focaliser sur l'analyse d'interfaces linguistiques. Parmi ces interfaces, nous avons présenté et illustré celle mettant en jeu le niveau pragmatique et le niveau prosodique. En nous appuyant sur une approche résolument modulaire, nous avons montré que les unités linguistiques relatives à ces niveaux peuvent être représentées sous la forme d'*objets informatiques*, qui ont chacun des attributs particuliers et qui sont reliés entre eux par des liens de dépendance.

Ce constat nous a menés à établir un lien transparent entre une vision modulaire des différents niveaux linguistiques et la méthode d'analyse informatique dite *orientée objet*. Il apparaît que les arborescences linguistiques que nous avons présentées peuvent se formaliser comme des objets, *instances* d'un nombre très réduit de *classes*, en relation les uns avec les autres. Ce rapprochement n'est pas que théorique, il a donné lieu ces dernières années au développement de OOPS, module Python qui permet d'analyser les interfaces linguistiques et qui prend en entrée des transcriptions annotées accompagnées de fichiers TextGrid contenant les temps de parole. Cet outil est diffusé librement à l'usage de la communauté en licence libre GPLv3.

Nous avons montré que l'approche objet présente de nombreux avantages. En premier lieu, elle s'harmonise avec un point de vue modulaire. Deuxièmement, elle permet de manipuler conjointement plusieurs hiérarchies provenant de niveaux linguistiques différents. Enfin, il est possible de rajouter de nouvelles arborescences linguistiques par le simple stockage de nouveaux objets. Nous avons illustré l'utilisation de OOPS par l'utilisation d'une méthode de stylisation automatique des contours prosodiques des unités illocutoires du corpus Rhapsodie, unités dont le support est la transcription, tandis qu'une analyse prosodique se fait à partir du signal de parole. Établir des ponts entre de telles unités est la vocation de OOPS, qui a été mis en œuvre très simplement dans ce but. Grâce à cette étude, nous avons montré par une étude systématique de plus de 3h de français parlé que dans le cadre de cette expérience très préliminaire, les idées de Delattre datant de 1966 sur l'intonation du français en termes de contours globaux semblent se vérifier en pratique sur le corpus que nous utilisons.

Références bibliographiques

- Aubergé, A. (1991), « La Synthèse de la Parole : “Des Règles aux Lexiques” », Thèse de doctorat, Université Pierre Mendès-France, Grenoble, France
- Austin, J-L. (1962), « How to Do Things With Words », Cambridge (Mass.)
- Avanzi, M., Obin, N., Lacheret, A. et Victorri, B. (2011), Vers une modélisation continue de la structure prosodique. Le cas des proéminences accentuelles, *Journal of French Languages Studies*, 1-21
- Avanzi, M., Lacheret, A. et Victorri, B. (2008), « Analor, un outil d'aide pour la modélisation de l'interface prosodie-grammaire », *Travaux linguistiques du CERLICO*, 27-46
- Bailly, V. et Aubergé, G. (1995), « Generation of intonation: a global approach », *Fourth European Conference on Speech Communication and Technology*
- Beliao, J., Lacheret, A. et Kahane, S. (2013), *Modéliser l'interface intonosyntaxique : ratio et synchronisation entre périodes intonatives et unités illocutoires*, Interface Discours Prosodie (IDP2013), Université de Leuven (KU Leuven), Leuven, Belgique
- Beliao, J., et Lacheret, A. (2013), *Disfluencies and discursive markers : when prosody and syntax plan discourse*, The 6th Workshop on Disfluency in Spontaneous Speech (DISS2013), KTH Royal Institute of Technology, Stockholm, Sweden
- Benzitoun, C., Dister, A., Gerdes, K., Kahane, S., Pietrandrea, P. et Sabio, F. (2010), Tu veux couper là faut dire pourquoi. propositions pour une segmentation syntaxique du français parlé, Actes du Congrès Mondial de Linguistique française, La Nouvelle Orléans.
- Benzitoun, C., Deulofeu, J., Kahane, S., Pietrandrea, P., Bolly, C., Debaisieux, J.-M., Dister, A., Lefeuvre, F., Rossi-Gensane, N., Sabio, F. et Tanguy, N. (à paraître), The macrosyntactic annotation, 6, *Benjamins*
- Benzitoun, C., Dister, A., Gerdes, K., Kahane, S. & Marlet, R. (2009), « annoter du des textes tu te demandes si c'est syntaxique tu vois », *The 28th Conference on Lexis and Grammar*, Arena Romanistica 4, Presses de l'Université de Bergen, 16-27
- Benzitoun, C. et Sabio, F. (en collaboration avec l'ensemble de l'équipe syntaxe Rhapsodie et en particulier Paola Pietrandrea & Sylvain Kahane) (2012), Protocole de codage macrosyntaxique
- Berrendonner, A., (2002), “ Morpho-syntaxe, pragma-syntaxe, et ambivalences sémantiques”, in Leth Andersen, Hanne et Henning Nølke, eds., *Macro-syntaxe et macro-sémantique*, Berne, Peter Lang, 23-41
- Blanche-Benveniste, C. (1990), Un modèle d'analyse syntaxique 'en grilles' pour les productions orales
Anuario de Psicologia Liliane Tolchinsky (coord.) Barcelona, vol. 47, 11-28
- Booch, G., et Rumbaugh, J. (1996). *The unified modeling language*. University Video Communications and the Association for Computing Machinery.
- Boersma, P. et Weenink, D. (2012), PRAAT - doing phonetics by computer
- Bonvino, E., Masini, F. et Pietrandrea, P. (2009), List Constructions: a semantic network
Troisième Conférence Internationale de l'AFLiCo, Nanterre

- Branca-Rosoff S., Fleury S., Lefevre Fl. et Pires, M. (2012), Discours sur la ville. Corpus de Français Parlé Parisien des années 2000 (CFPP2000)
- Campione, E., Hirst, D. et Véronis, J. (2000), Automatic Stylisation and Symbolic Coding of F0 : Implementations of the INTSINT Model. Dordrecht : Kluwer, ch. Intonation. Research and Applications
- Durand, J., Laks, B. & Lyche, C. (2009), Le projet PFC (phonologie du français contemporain): une source de données primaires structurées, in : J. Durand, B. Laks & C. Lyche (eds.), Phonologie, variation et accents du français. Hermès, Paris, 19-61.
- Gerdes, K. et Kahane, S. (2009), « Speaking In Piles: Paradigmatic Annotation Of French Spoken Corpus », *Proceedings of the Fifth Corpus Linguistics Conference, Liverpool.*
- Gerdes, K. (2012), Arborator : A tool for collaborative dependency annotation, <http://arborator.ilpqa.fr/>
- Goldman, J.-P. (2011), EasyAlign: an automatic phonetic alignment tool under Praat InterSpeech
- Hamilton, M. (1999), Software Development: A Guide to Building Reliable Systems p.48
- Kahane, S. et Pietrandrea, P. (2012), Typologie des entassements en français, Actes de la conférence Linx
- Lacheret, Kahane, Pietrandrea (Eds.), (à par 2015), Rhapsodie: a Prosodic and Syntactic Treebank for Spoken French, Amsterdam, Benjamins
- Lacheret A. Kahane S. Pietrandrea P, Avanzi M., Victorri B. (2011), « Oui mais elle est où la coupure là ? Quand syntaxe et prosodie
- Lacheret, A. et Victorri, B. (2002), La période intonative comme unité d'analyse pour l'étude du français parlé : modélisation prosodique et enjeux linguistiques, *Verbum*, 55-72
- Mertens, P. (2004), "The Prosogram : Semi-Automatic Transcription of Prosody based on a Tonal Perception Model," in *Speech Prosody*, Nara, Japan, pp. 549–552.
- Mertens P. (1987) , L'intonation du français : de la description linguistique à la reconnaissance automatique, Thèse de Doctorat, Université de Louvain
- Obin, N., Beliao, J., Veaux, C., Lacheret, A. (soumis) SLAM: Automatic Stylization and Labelling of Speech Melody, *SpeechProsody*, Dublin
- Rumbaugh, J. (1990), Object-Oriented Modeling and Design
- Rumbaugh, J., Jacobson, I., & Booch, G. (1999). The unified modeling language reference manual.
- Shriberg, E., Stolcke, A., Jurafsky, D., Coccaro, N., Meteer, M., Bates, R., et Van Ess-Dykema, C. (1998), Can prosody aid the automatic classification of dialog acts in conversational speech?. *Language and speech*, 41(3-4), 443-492
- Searle, John R. (1975), « « A Taxonomy of Illocutionary Acts », in: Günderson, K. (ed.), *Language, Mind, and Knowledge, (Minneapolis Studies in the Philosophy of Science, vol. 7)*, University of Minneapolis Press, p. 344-69
- Searle, John R. (1969), « Speech Acts », Cambridge University Press
- Syrdal, K., Hirschberg, A.-J., McGory, J., et Beckman, M. (2001), "Automatic ToBI Prediction and Alignment to Speed Manual Labeling of Prosody," *Speech Communication*, vol. 33, no. 1-2, pp. 135–151

ⁱ L'ensemble du corpus peut-être téléchargé et consulté sur le site du projet Rhapsodie : <http://www.projet-rhapsodie.fr/>

ⁱⁱ Les membres du projet considèrent les unités illocutoires comme partie intégrante de la macrosyntaxe. Nous considérons ici dans le cadre de notre l'approche objet qu'elles constituent un niveau linguistique à part entière en tant qu'unités pragmatiques.

ⁱⁱⁱ On pourra trouver le détails des symboles du balisage manuel du corpus Rhapsodie dans (Benzitoun et Sabio 2012, document disponible sur le site du projet Rhapsodie).

^{iv} On remarque ici que parfois le même symbole du balisage encode plusieurs phénomènes en même temps, par exemple // marque à la fois une unité macrosyntaxique englobante et une unité illocutoire.

^v Un contour mélodique est représenté par ses valeurs initiales et finales, puis par le type de saillance ainsi que sa position si elle est suffisamment prononcée. L'alphabet de chaque symbole (initial, saillance, final) est composé de 5 registres élémentaires, relatifs au locuteur.