Extraction de données orales multi-annotées

Brigitte Bigi[1] Tatsuya Watanabe[1,2]
(1) Laboratoire Parole et Langage, AMU, CNRS, 5 avenue Pasteur, 13100 Aix-en-Provence
(2) Ortolang
brigitte.bigi@lpl-aix.fr, tatsuya.watanabe@atilf.fr

Résumé. Cet article aborde le problème de l'extraction de données orales multi-annotées : nous proposons une solution intermédiaire, entre d'une part les systèmes de requêtages très évolués mais qui nécessitent des données structurées, d'autre part les données (multi-)annotées des utilisateurs qui sont hétérogènes. Notre proposition s'appuie sur 2 fonctions principales : une fonction booléenne pour filtrer sur le contenu, et une fonction de relation qui implémente l'algèbre de Allen. Le principal avantage de cette approche réside dans sa généricité : le fonctionnement sera identique que les annotations proviennent de Praat, Transcriber, Elan ou tout autre logiciel d'annotation. De plus, deux niveaux d'utilisation ont été développés : une interface graphique qui ne nécessite aucune compétence ou connaissance spécifique de la part de l'utilisateur, et un interrogation par scripts en langage Python. L'approche a été implémentée dans le logiciel SPPAS, distribué sous licence GPL.

Abstract. This paper addresses the problem of extracting multimodal annotated data in the linguistic field ranging from general linguistic to domain specific information. Our proposal can be considered as a solution or a least an intermediary solution that can link together requesting systems and expert data from various annotation tools. The system is partly based on the Allen algebra and consists in creating filters based on two functions : a boolean function and a relation function. The main advantage of this approach lies in its genericity : it will work identically with annotations from Praat, Transcriber, Elan or from any other annotation software. Furthermore, two levels of usage have been developed : a graphical user interface graph that not requires any skill or knowledge, and a query form in Python. This system is included in the software SPPAS and is distributed under the terms of the GPL license.

Mots-clés : multimodalité, corpus, extraction.

Keywords: multimodality, corpus, extraction.

1 Introduction

Au cours de ces dix dernières années, la quantité de données linguistiques qui sont devenues disponibles a considérablement augmenté, non seulement pour les corpus écrits, mais aussi pour les données orales. Dans divers domaines de la linguistique, il est aujourd'hui de plus en plus attendu pour les linguistes que leurs études portent sur des quantités importantes de données empiriques, pouvant comprendre jusqu'à plusieurs heures de parole. Jusqu'à il y a quelques années, il était difficile, voire impossible, de manipuler de telles quantités de données. De nombreux logiciels, distribués librement, permettent désormais d'annoter manuellement des corpus oraux, tels que *Transcriber* (Barras *et al.*, 1998), *Praat* (Boersma, 2001), ou *WaveSurfer* (Sjölander et Beskow, 2000), pour ne citer que quelques uns des plus populaires (voir (Llisterri, 2011) pour une étude complète des logiciels d'analyse de la parole). D'autres outils sont plus orientés vers l'annotation manuelle à partir de vidéos, tels que *Elan* (Brugman *et al.*, 2004) ou *Anvil* (Kipp, 2013), mais on ne peut pas les utiliser pour effectuer des annotations phonétiques ou prosodiques précises. D'autres outils, ou boîtes à outils, permettent d'annoter automatiquement de très grandes quantités de données orales, pour divers niveaux d'annotations. Le LPL, par exemple, distribue le logiciel *SPPAS* (Bigi et Hirst, 2012), sous licence GPL, qui permet d'obtenir une segmentation automatique en mots, syllabes et phonèmes, à partir d'enregistrements audio transcrits. Un analyseur morpho-syntaxique y est également développé (Rauzy et Blache, 2009) et permet d'obtenir automatiquement un étiquetage de données orales alignées sur le signal.

La plupart du temps, les données annotées sont représentées sous la forme de "tiers". Ces "tiers" sont composées de séries

d'intervalles (ou de points pour le logiciel Praat). Les intervalles sont définis par un temps de début, un temps de fin et un label. Dans Praat, aucun lien ne peut être défini entre les "tiers". Dans ANVIL, par contre, avant toute utilisation, un schéma d'annotation doit être défini. De cette manière, des dépendances peuvent être établies entre les "tiers".

Ainsi, les annotations depuis ces dernières années ont ceci de particulier qu'elles concernent de plus en plus souvent différents domaines, notamment grâce aux développements logiciels récents. C'est ce qui est constaté dans (O'Halloran et Smith, 2012) : "in recent decades the rapid increase in sophistication and availability of technological (particularly computational) resources and techniques for analysis of multimodal text has no doubt driven the rapid increase in multimodal analyses appearing within a range of disciplines, vastly improving, as technology did for the study of speech earlier, our access to and understanding of multimodal text using, for example, multimodal annotation software".

Les annotations multimodales comprennent des informations qui peuvent concerner divers domaines tels que la prosodie, la phonétique, la syntaxe, le discours, la gestuelle, etc, comme par exemple dans (Blache *et al.*, 2010). Ainsi, le plus grand obstacle aujourd'hui auquel les linguistes doivent faire face n'est pas le stockage de données, ni leur annotation, mais plutôt leur *exploration*. Ce challenge est notamment mentionné dans (O'Halloran, 2011), "The major challenge to Multimodal Discourse Analysis is managing the detail and complexity involved in annotating, analysing, searching and retrieving multimodal semantics patterns within and across complex multimodal phenomena.".

Parallèlement, les moyens pour requêter des données se sont considérablement améliorés, passant de simples bases de données sous formes de tables interrogées à l'aide de langages comme SQL, à des systèmes complexes de stockage de l'information (OWL par exemple) permettant des requêtes élaborées à l'aide de langages (sparql ou Xquery par exemple). Cependant, à notre connaissance, aucun de ces systèmes ne permet l'exploration et l'extraction directement à partir des données issues des logiciels d'annotation.

Dans cet article, nous proposons une solution simple et originale qui permet d'explorer/extraire directement les données multi-annotées des utilisateurs. Le système proposé se situe à mi-distance entre d'une part les systèmes d'interrogations qui ne peuvent opérer que sur des données structurées et les recherches "élémentaires" à base de patrons que l'on trouve dans les différents logiciels d'annotations. Notre système consiste à créer des filtres, puis à les appliquer sur les données annotées. En sortie, il produit une nouvelle annotation en fonction de la sélection effectuée. La version actuelle permet de traiter des données provenant des logiciels d'annotation Transcriber, Praat, Elan ou à partir de fichiers CSV. De plus, d'autres formats pourront s'ajouter à cette liste sans remettre en cause le fonctionnement interne du système.

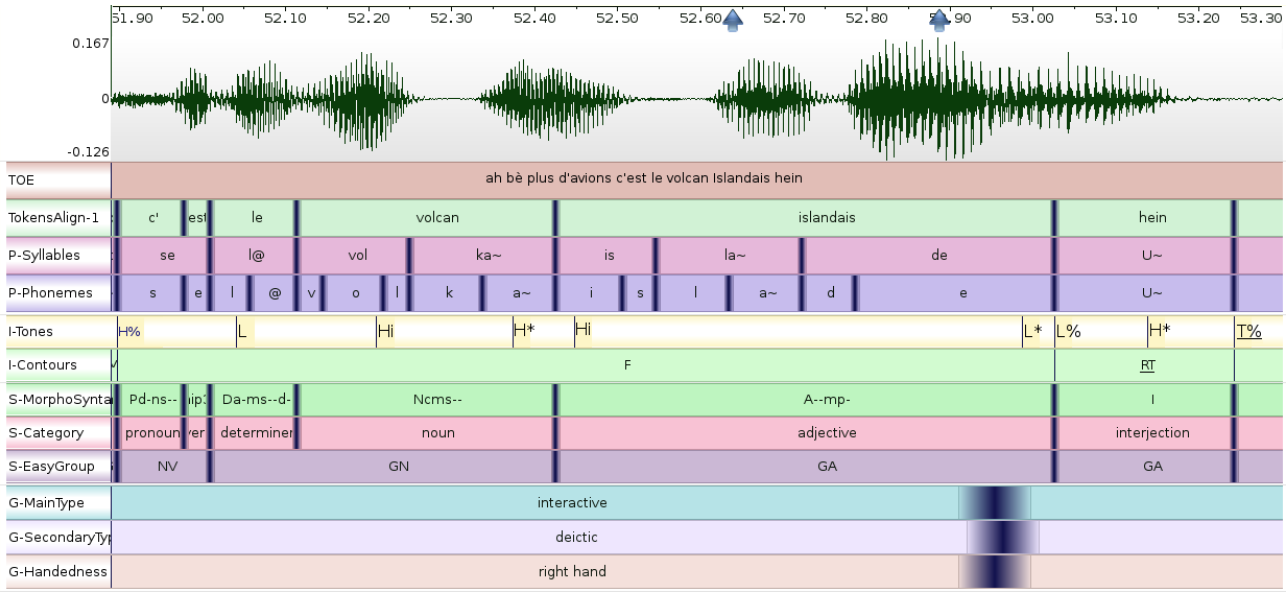


FIGURE 1 – Exemple de données multi-annotées manuellement et automatiquement, importées de 3 logiciels différents.

2 Représentation des données multi-annotées

La difficulté majeure dans le traitement de ces données provient essentiellement de la nature même des annotations : selon le type d'annotations, les bornes des intervalles sont placées de façon plus ou moins précise. Par exemple, lorsqu'il s'agit d'annotations phonétiques ou prosodiques la précision est de l'ordre de quelques millisecondes. D'autres annotations, comme celle des types de gestes, ne peuvent pas être plus précises que la durée d'une image, soit 40 ms la plupart du temps. Pour surmonter cette difficulté, une solution pour représenter les données a été proposée dans (Bigi *et al.*, 2014). Elle consiste à représenter un point dans le temps X à l'aide de deux valeurs : un "midpoint" M_x (valeur centrale) et un "radius" R_x (zone d'incertitude), tel que $X = (M_x, R_x)$.

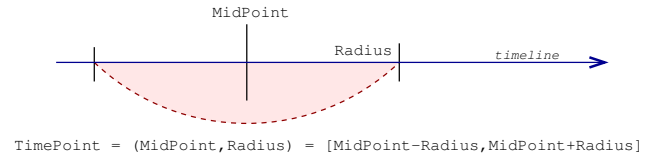


FIGURE 2 – Représentation du temps des données annotées

L'utilisation la plus simple de ces valeurs consiste à attribuer au radius une valeur égale à la moitié de la valeur du taux d'échantillonnage du média annoté (par exemple, 20 ms dans le cas d'une vidéo), ou à la durée que représente 1 pixel à l'écran au moment de l'annotation. Les comparaisons de deux points dans le temps X et Y se feront comme décrit dans l'équation 1.

$$\begin{aligned} X = Y &\Leftrightarrow |M_X - M_Y| \leq R_X + R_Y \\ X < Y &\Leftrightarrow \neg(X = Y) \wedge (M_X < M_Y) \\ X > Y &\Leftrightarrow \neg(X = Y) \wedge (M_X > M_Y) \end{aligned} \quad (1)$$

La figure 1 montre différentes annotations dans lesquelles les bornes utilisent cette notion. La valeur de radius a été fixée à 30 ms pour les annotations gestuelles, 0 ms pour les annotations prosodiques, et 5 ms pour les autres annotations.

Le système d'interrogation proposé dans cet article s'appuie sur cette représentation des données afin de rendre l'extraction plus robuste (en particulier lorsqu'il sera question de faire des extractions multi-domaines).

3 Filtrage sur un seul niveau d'annotation : fonction booléenne

Cette section décrit l'ensemble des filtres qui ont été prévus pour extraire les données d'une seule "tier" (une ligne d'annotation). Nous montrons la syntaxe que nous proposons d'utiliser avec **l'interpréteur Python**. Chacune des fonctionnalités peut être utilisée via **l'interface graphique** (figure 3). Dans la suite, nous noterons $Bool$ une fonction booléenne qui s'applique sur chacune des annotations d'une tier T, à l'aide d'un filtre f.

La recherche à base de patrons constitue une partie importante de tout système d'extraction. Ainsi, les filtres qui suivent sont proposés afin de sélectionner les annotations selon leur label (on note P, le patron à chercher) :

- correspondance exacte : $f = T(Bool(exact = P))$,
- contient : $f = T(Bool(contains = P))$,
- commence par, $f = T(Bool(startswith = P))$,
- termine par, $f = T(Bool(endswith = P))$,
- expression régulière, $f = T(Bool(regexp = R))$.

Les fonctions booléennes peuvent être inversées (si elles sont précédées par le caractère $\sim$). De plus, les recherches peuvent être sensibles à la casse ou non (sauf le dernier), si le critère commence par 'i', par exemple : $f = T(\sim Bool(iexact = P))$.

Deux types de filtres peuvent être créés avec des critères de temps : en fonction de la durée d'une annotation, ou de ses valeurs de début et de fin :

- une durée inférieure à une valeur v : $f = T(Bool(duration\_lt = v))$,
- une durée inférieure ou égale à une valeur v :$f = T(Bool(duration\_le = v))$,
- une durée supérieure à une valeur v : $f = T(Bool(duration\_gt = v))$,

— une durée supérieure ou égale à une valeur v : $f = T(Bool(duration\_ge = v))$,
— une durée égale à une valeur v : $f = T(Bool(duration\_e = v))$,
— le début doit être après ou égale à un temps t : $f = T(Bool(begin\_ge = t))$,
— la fin doit être avant ou égale à un temps t : $f = T(Bool(end\_le = t))$.

L'implémentation des opérateurs & ("ET") et | ("OU") permet de combiner ces critères. Par exemple, on pourra écrire : $f = T(((Bool(exact = P_1) \mid Bool(exact = P_2))) \ \& \ Bool(duration\_lt = v))$.

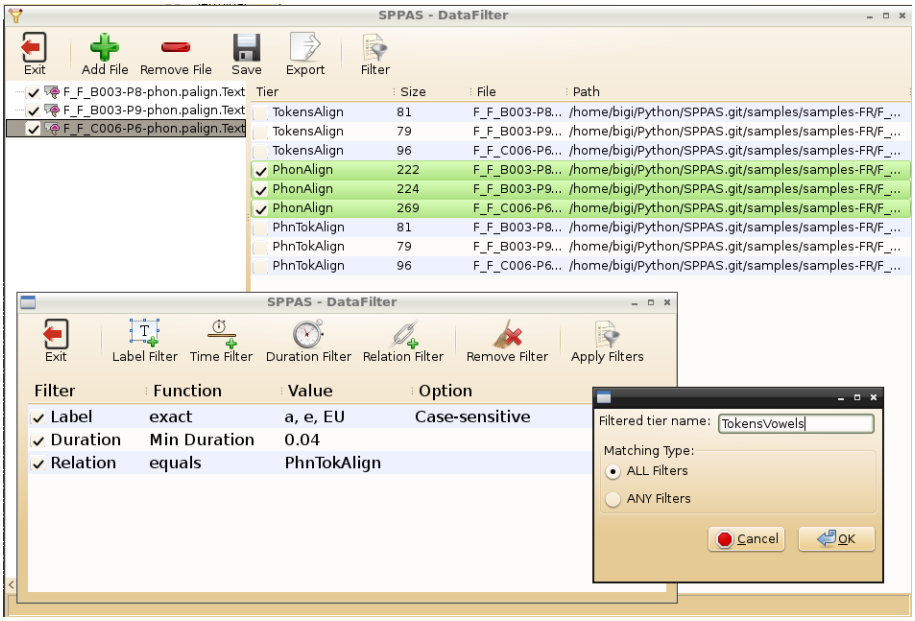


FIGURE 3 – Filtrage à l'aide de l'interface graphique.

4 Filtrage entre différents niveaux d'annotations : fonction relation

Dans la théorie de Allen, les manières d'ordonner les extrémités de deux intervalles sont décrites par 13 relations atomiques (before, after, meets, met by, overlaps, overlapped by, starts, started by, finishes, finished by, contains, during, equals). L'implémentation que nous proposons des relations de Allen est décrite dans la table 1, où l'on note $X = [X^-, X^+]$ et $Y = [Y^-, Y^+]$ deux intervalles non réduits à un point. Nous avons étendue notre implémentation aux 25 relations du modèle INDU ("INterval and DUration"), proposé dans (Pujari *et al.*, 2000), qui intègre des contraintes de durées des intervalles aux relations de Allen : X est de durée inférieure, supérieure ou égale à Y.

De plus, nous avons ajouté des critères de délai pour les relations "after" et "before", ainsi que pour les relations "overlaps" et "overlapped by".

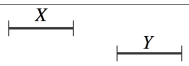
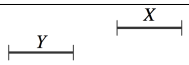
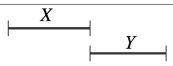
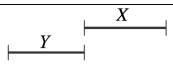
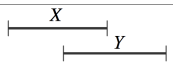
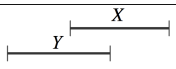
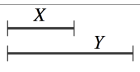
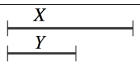
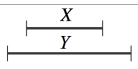
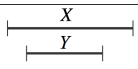
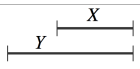
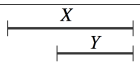
Pour simplifier l'utilisation, nous proposons 3 meta-règles : "equals", "convergent" et "disjoint". La meta-relation convergent inclue overlaps, overlapped by, starts, started by, during, contains, finishes et finished by. La meta-relation "disjoint" inclue before, after, meets et met by.

Enfin, dans la mesure où certains logiciels proposent d'utiliser des annotations sous forme de points plutôt que d'intervalles, nous avons implémenté cette possibilité.

5 Exemple de requête

À titre d'illustration, cette section présente l'implémentation Python d'une requête, sachant que celle-ci peut tout aussi bien être réalisée avec l'interface graphique : **Quels sont les gestes pendant les pauses ?**

TABLE 1 – Relations de Allen entre deux intervalles, et leur implémentation à l'aide de deux filtres f_1 et f_2

Filtre relationnel	Illustration	Description
$f_1.Link(f_2, Rel("before"))$		$X^+ < Y^-)$
$f_1.Link(f_2, Rel("after"))$		$(X^- > Y^+)$
$f_1.Link(f_2, Rel("meets"))$		$(X^+ = Y^-)$
$f_1.Link(f_2, Rel("metby"))$		$(X^- = Y^+)$
$f_1.Link(f_2, Rel("overlaps"))$		$(X^- < Y^-) \wedge (X^+ > Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("overlappedby"))$		$(X^- > Y^-) \wedge (X^- < Y^+) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("starts"))$		$(X^- = Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("startedby"))$		$(X^- = Y^-) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("during"))$		$(X^- > Y^-) \wedge (X^+ < Y^+)$
$f_1.Link(f_2, Rel("contains"))$		$(X^- < Y^-) \wedge (X^+ > Y^+)$
$f_1.Link(f_2, Rel("finishes"))$		$(X^- > Y^-) \wedge (X^+ = Y^+)$
$f_1.Link(f_2, Rel("finishedby"))$		$(X^- < Y^-) \wedge (X^+ = Y^+)$
$f_1.Link(f_2, Rel("equals"))$		$(X^- = Y^-) \wedge (X^+ = Y^+)$

Après avoir récupéré les tiers concernées par la recherche, il faut fixer une valeur de radius appropriée, qui dépend du type d'annotation. Cette étape favorisera les comparaisons, en s'appuyant sur les equations décrites en (1), dans les relations de la table 1. La valeur de temps s'indique en secondes :

```
gestures.SetRadius(0.02)
tokens.SetRadius(0.001)
```

Ensuite, une fonction booléenne est créée pour chacune des tiers. Dans cet exemple, il n'y a pas de critère de sélection sur les gestes. Par contre, il faut sélectionner les pauses dans la tier des tokens. Nous garderons les pauses silencieuses de plus de 200ms et les pauses pleines de plus de 100ms. Ces fonctions booléennes sont données en paramètre pour créer les filtres :

```
fgest = gestures()
ftokens = tokens((Bool(exact='euh') & Bool(duration_gt=0.1)) |\
          (Bool(exact='#') & Bool(duration_gt=0.2)))
```

Il faut alors déclarer une fonction de relations, qui dresse l'inventaire de chacune des relations de Allen qui pourra être vraie pour qu'une annotation soit acceptée :

```
relation = Rel('starts') | Rel('startedby') | Rel('finishes') | Rel('finishedby') |\
           Rel('during') | Rel('contains') | Rel('overlaps') | Rel('overlappedby')
```

La fonction de relation sert à créer un nouveau filtre pour obtenir une tier qui ne contient que les gestes recherchés :

```
f = fgest.Link(ftokens, relation)
new_gest_tier = fget.Filter()
```

6 Conclusion

Nous avons proposé un système d'extraction de données multi-annotées provenant, éventuellement, de différents logiciels d'annotation couramment utilisés par les linguistes. Ce système repose que l'implémentation de deux fonctions : une fonction booléenne pour filtrer sur le contenu des annotations et une fonction de relation basée sur l'algèbre de Allen. L'avantage principal de notre proposition réside dans le fait que *les requêtes peuvent être formulées par les annotateurs eux-mêmes, directement sur leurs annotations*, notamment via l'interface graphique. Ce système permet ainsi de répondre concrètement à un besoin exprimé par les annotateurs, qui sont dans l'impossibilité d'utiliser les systèmes de requêtes existant (SQL, Xquery, etc). Le système proposé fait parti de SPPAS, logiciel librement distribué. De plus, avec des compétences élémentaires en Python, les données peuvent également être interrogées à l'aide de l'API fournie. Dans un futur proche, il est notamment prévu d'ajouter l'importation d'annotations provenant d'autres logiciels.

Remerciements

Ce travail, mené dans le cadre de la mise en place de l'Equipex ORTOLANG a bénéficié d'une aide de l'État gérée par l'ANR au titre du programme Investissement d'Avenir portant la référence ANR-11-EQPX0032.

Références

BARRAS, C., GEOFFROIS, E., WU, Z. et LIBERMAN, M. (1998). Transcriber : a free tool for segmenting, labeling and transcribing speech. *In First International Conference on Language Resources and Evaluation*, pages 1373–1376, Granada (Spain).

BIGI, B. et HIRST, D. (2012). SPeech Phonetization Alignment and Syllabification (SPPAS) : a tool for the automatic analysis of speech prosody. *In* TONGJI UNIVERSITY PRESS, I., éditeur : *Speech Prosody*, Shanghai (China).

BIGI, B., WATANABE, T. et PRÉVOT, L. (2014). Representing multimodal linguistics annotated data. *In Language Resources and Evaluation Conference*, Reykjavik (Iceland).

BLACHE, P., BERTRAND, R., BIGI, B., BRUNO, E., CELA, E., ESPESSER, R., FERRÉ, G., GUARDIOLA, M., HIRST, D., MAGRO, E.-P., MARTIN, J.-C., MEUNIER, C., MOREL, M.-A., MURISASCO, E., NESTERENKO, I., NOCERA, P., PALLAUD, B., PRÉVOT, L., PRIEGO-VALVERDE, B., SEINTURIER, J., TAN, N., TELLIER, M. et RAUZY, S. (2010). Multimodal annotation of conversational data. *In The Fourth Linguistic Annotation Workshop*, pages 186–191, Uppsala (Sueden).

BOERSMA, P. (2001). Praat, a system for doing phonetics by computer. *Glot International*, 5 :9/10:341–345.

BRUGMAN, H., RUSSEL, A. et NIJMEGEN, X. (2004). Annotating multi-media / multimodal resources with ELAN. *In Fourth International Conference on Language Resources and Evaluation*, pages 2065–2068, Lisbon (Portugal).

KIPP, M. (2013). *ANVIL : A Universal Video Research Tool*. J. Durand, U. Gut, G. Kristofferson (Hrsg.) Handbook of Corpus Phonology, Oxford University Press.

LLISTERRI, J. (2011). Speech analysis and transcription software. http ://liceu.uab.es/˜joaquim/phonetics/fon_anal_acus/herram_anal_acus.html.

O'HALLORAN, K. L. (2011). Multimodal discourse analysis. *Companion to Discourse*, pages 120–137.

O'HALLORAN, K. L. et SMITH, B. A. (2012). Multimodal text analysis. *The Encyclopedia of Applied Linguistics.*

PUJARI, A. K., KUMARI, G. V. et SATTAR, A. (2000). INDU : An interval duration network. *In Sixteenth Australian joint conference on AI*, pages 291–303. SpringerVerlag.

RAUZY, S. et BLACHE, P. (2009). Un point sur les outils du LPL pour l'analyse syntaxique du français. *In Actes de la Journée ATALA "Quels analyseurs syntaxiques pour le français ?"*, Paris (France).

SJÖLANDER, K. et BESKOW, J. (2000). WAVESURFER - an open source speech tool. *In 6th International Conference of Spoken Language Processing*, Beijing (China).