Découverte non supervisée de lexique à partir d'un corpus multimodal pour la documentation des langues en danger

William N Havard

Découverte non supervisée de lexique à partir d'un corpus multimodal pour la documentation des langues en danger

William N. Havard

Sous la direction de Laurent Besacier

Tuteur : Olivier Kraif

Laboratoire d'informatique de Grenoble — LIG

UFR LLASIC

Département Informatique intégrée en Langues, Lettres et Langage (I3L)

Mémoire de Master 2 mention Sciences du langage – 20 crédits

Parcours : Industries de la langue

Année universitaire 2016-2017

Découverte non supervisée de lexique à partir d'un corpus multimodal pour la documentation des langues en danger

William N. Havard

Sous la direction de Laurent Besacier

Tuteur : Olivier Kraif

Laboratoire d'informatique de Grenoble — LIG

UFR LLASIC

Département Informatique intégrée en Langues, Lettres et Langage (I3L)

Mémoire de Master 2 mention Sciences du langage – 20 crédits

Parcours : Industries de la langue

Année universitaire 2016-2017

REMERCIEMENTS

Je tiens tout d'abord à adresser mes remerciements les plus sincères à Monsieur Laurent Besacier pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail de recherche, pour son accompagnement tout au long du stage et de la rédaction de ce mémoire ainsi que pour ses nombreux conseils.

Je souhaiterais également remercier Monsieur Olivier Kraif pour ses encouragements ainsi que ses conseils lors la rédaction de ce mémoire.

Je remercie également Monsieur Georges Antoniadis et Monsieur Jean-Pierre Chevrot d'avoir accepté de faire partie de mon jury. Je tiens également à remercier Monsieur Chevrot de l'intérêt qu'il a porté a mon travail de recherche et pour ses remarques qui ont donné de nouvelles perspectives à mon travail.

Je souhaiterais également remercier Monsieur Guy-Noël Kouarata et Madame Annie Rialland de m'avoir permis d'utiliser leur corpus en Mboshi. Je remercie également Monsieur Dimmendaal, Madame Schneider-Blum et Madame Hellwig de m'avoir permis d'utiliser les images qu'ils ont prises lors de la constitution de leurs corpus en Tima et Tabaq. Finalement je souhaiterais remercier les équipes de Voxygen de m'avoir permis d'utiliser leur système de synthèse vocale.

Je tiens également à remercier l'ensemble de l'équipe pédagogique du Master Industries de la langue pour leur disponibilité et pour avoir toujours transmis leurs connaissances avec passion.

Je n'oublie évidemment pas mes camarades de classe. Merci pour la bonne ambiance et l'esprit de travail qui ont régné pendant ces deux années. Je tenais tout particulièrement à remercier Anne-Laure, Ali Can, Doriane, Ieva, Louise, Pauline, Renaud et Sylvain pour leur soutien et encouragements.

Je souhaite également remercier Lauren ainsi que mes amis de Corps-Nuds, tout particulièrement Marion, Océane et Rodolphe.

Je souhaiterais remercier ma famille en Angleterre, mon père Christopher, Jane, mon frère et ma sœur Jack et Keerah, ainsi que mes grands-parents Jean et Tony.

Last but not least, je souhaiterais très sincèrement remercier ma mère, Marie Odile, mon grand-père, Joseph, et ma grand-mère Marie, pour leur soutien et accompagnement tout au long de mes années d'études. Je ne serais pas arrivé là où j'en suis s'ils n'avaient pas été là.

Sommaire

Introduction

Ce mémoire de recherche a pour but d'explorer les possibilités offertes par le traitement automatique des langues pour la documentation des langues en danger. Ce mémoire fait suite à un stage de recherche effectué dans le cadre du Master 2 en Sciences du langage, parcours Industries de la langue proposé par l'Université Grenoble Alpes. Notre travail de recherche a été encadré par M. Laurent Besacier, Professeur à l'Université Grenoble Alpes et directeur de l'équipe GETALP du LIG, ainsi que par M. Olivier Kraif, Maitre de conférences (HDR) à l'Université Grenoble Alpes. Notre stage s'est déroulé au Laboratoire d'Informatique de Grenoble (LIG) dans l'équipe GETALP (Groupe d'Étude en Traduction Automatique/Traitement Automatisé des Langues et de la Parole). Notre travail de recherche s'inscrit dans deux des axes de recherche de cette équipe, à savoir le traitement de la parole ainsi que le traitement des langues peu dotées.

Nous présenterons dans cette introduction le contexte dans lequel se situe notre sujet de mémoire puis nous en détaillerons la structure.

Les linguistes estiment qu'il existe environ 7.000 langues aujourd'hui parlées dans le monde, et que près de 50% d'entre elles – voire même 90% pour les plus alarmistes – pourraient disparaître d'ici à la fin du siècle (Austin et al., 2011, pp. 1–2). P. Austin et J. Sallabank précisent que « la disparition des langues n'est pas un phénomène nouveau, mais l'accélération de la disparition de celles-ci l'est » (Austin et al., 2011, p. 33). Il est donc urgent d'agir afin de préserver les langues qui risquent de disparaître. Certaines langues présentent des particularités uniques qui nous permettraient peut-être de mieux comprendre le langage. La perte d'une langue signifie également la perte d'une culture, avec ses modes de pensées, ses histoires et légendes. Il est ainsi essentiel de sauvegarder les langues afin que ce savoir ne se perde jamais. La perte d'une langue peut parfois être vécue comme un grand traumatisme par ses locuteurs comme en atteste la citation suivante en judéo-espagnol de l'écrivain Français Marcel Cohen :

> *« Kyero eskrivirte en djudyo antes ke no keda nada del avlar de mis padres. No saves, Antonio, lo ke es morirse en su lingua. Es komo kedarse soliko en el silensyo kada dya ke Dyo da, komo ser sikileoso sin saver porke […] Kale ser loko para pensar ke, en eyas, podryas, ser un dya el mousafir de ti mizmo.» (Marcel Cohen)*
>
> *« Je voulais t'écrire en djudyo avant que s'éteigne tout à fait la langue de mes ancêtres. Tu n'imagines pas, Antonio, ce qu'est l'agonie d'une langue. C'est un peu comme se retrouver seul dans le silence. C'est se sentir sikileoso* (NdT anxieux, oppressé) *sans comprendre pourquoi. […] Comment imaginer que nous puissions devenir un jour, dans notre propre langue, les mousafires* (NdT étranger, visiteur) *de nous-mêmes ?»* (Cohen, 1997)

Le but de notre stage de recherche consiste à explorer les possibilités offertes par un système de découverte non supervisée de lexique à partir d'un corpus multimodal – incluant de la parole et des images – pour documenter une langue en danger.

La découverte non supervisée de lexique a pour but de découvrir automatiquement des unités lexicales à partir du signal de parole. Ainsi, le signal de parole est notre seule ressource et nous ne disposons pas d'autres informations – comme des transcriptions par exemple – qui permettraient de nous aider. En effet, une telle contrainte correspond à des scénarios réels puisque les langues en danger ne disposent généralement pas d'une écriture et ne sont généralement pas décrites. Ainsi, il nous est impossible d'utiliser des systèmes de reconnaissance de parole standards puisque ceux-ci nécessitent de nombreuses ressources telles que des modèles de langue, des modèles acoustiques et des dictionnaires phonétisés.

Les unités lexicales découvertes ne correspondent pas forcément à des mots comme nous le verrons, mais peuvent correspondre à des unités plus grandes, des *chunks* par exemple ou des bigrammes fréquents. Une fois ces unités lexicales découvertes, il s'agit ensuite d'établir des méthodes afin de pouvoir comprendre la signification de ces unités – éventuellement à la lumière du contenu des images associées aux enregistrements.

L'utilisation d'un corpus multimodal pour notre recherche a été motivée par le fait que nous disposions d'un tel corpus dans une langue en danger : le Mboshi. Les signaux de parole qui le constituent ont été obtenus par élicitation par le biais d'images. Ainsi, la parole obtenue fait directement référence au contenu des images. A ce jour, l'utilisation d'un système de découverte non supervisée de lexique et l'utilisation d'un corpus multimodal incluant parole et image pour documenter une vraie langue en danger n'a jamais été entreprise. En effet, les systèmes de découverte non supervisée de lexique ont été appliqués à des langues bien dotées pour lesquelles leurs structures étaient connues. De tels systèmes ont par exemple été mis en œuvre pour comprendre l'acquisition du langage chez les enfants (Roy and Pentland, 2002), (Johnson, 2009) et (Goldwater, 2006). Ainsi, appliquer un système de découverte non supervisée de lexique à une langue en danger permet de tester ses performances sur des langues inédites et permet d'évaluer sa capacité à généraliser.

Nous avons fait dans le **Chapitre 1** une revue de la littérature qui nous a permis d'identifier les méthodes employées par les linguistes de terrain afin de documenter une langue en danger, et ainsi voir leurs avantages et limites. Cela nous a également permis de voir dans quelle mesure les nouvelles technologies sont utilisées par les linguistes de terrain afin de les aider dans leur tâche. Ce chapitre nous a également permis d'identifier les méthodes employées pour découvrir du lexique de

manière non supervisée, que ce soit pour la documentation des langues ou pour modéliser l'acquisition du langage. Finalement, nous avons fait un parallèle entre les langues en danger et les langues peu dotées pour comprendre si les technologies appliquées à ces dernières sont aussi applicables aux langues en danger.

Dans le **Chapitre 2**, nous décrivons la méthodologie que nous avons employée afin de construire un corpus multimodal de grande taille. En effet, cela s'imposait puisque notre corpus en Mboshi est très petit et ne dispose d'aucune transcription ou traduction. Ainsi, il était nécessaire de construire un corpus pour nous permettre d'évaluer les approches computationnelles en matière de découverte non supervisée de lexique.

Le **Chapitre 3** décrit le fonctionnement d'un système de découverte non supervisée de lexique : ZRTools. Nous avons évalué les performances de ce système sur le corpus que nous avons construit. Cela nous a permis de dégager les paramètres optimums et nous a également permis de voir les défauts d'un tel système. Nous terminons ce chapitre en appliquant le système de découverte non supervisée de lexique à une vraie langue en danger : le Mboshi.

Ainsi, notre travail de recherche s'inscrit dans un contexte large, à la croisée de plusieurs domaines : la documentation des langues, la psycholinguistique avec la modélisation de l'acquisition du langage et finalement la linguistique-informatique et l'informatique avec l'utilisation de méthodes non supervisées.

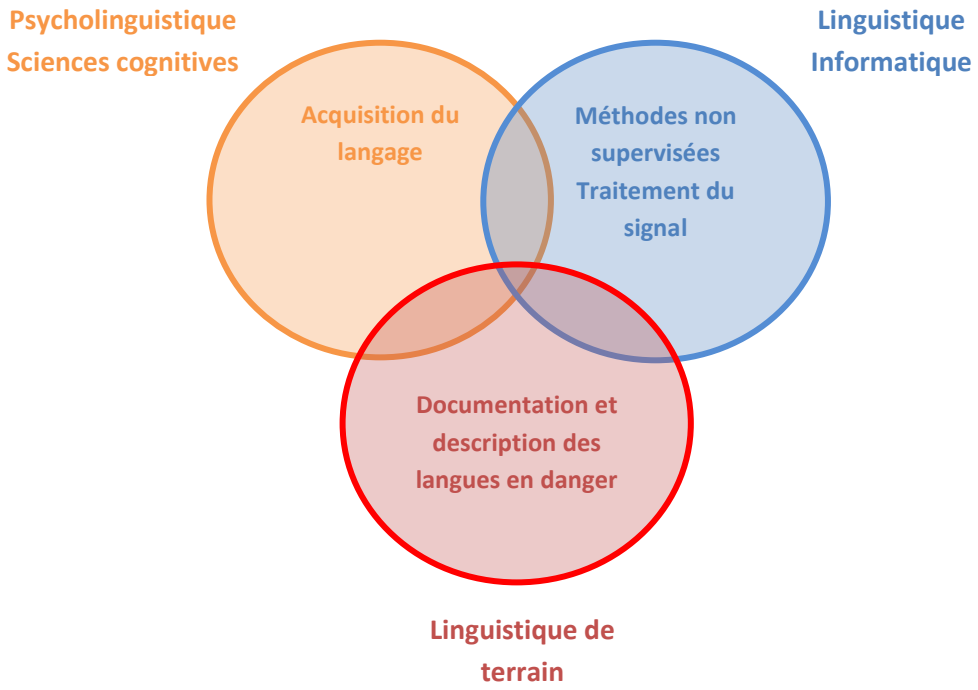


Figure 1 Champs disciplinaires dans lesquels se situe notre travail

Chapitre 1

–

État de l’art

Notre travail de recherche consistera à regarder comment le traitement automatique des langues peut aider à la préservation des langues en danger, en particulier du point de vue de la documentation et encore plus précisément du lexique. Dans un premier temps, nous regardons ce que recouvre exactement la notion de « documentation d'une langue en danger ». Par la suite, nous nous intéresserons aux travaux permettant la découverte non supervisée de lexique. Pour terminer, nous ferons un point sur les corpus existants.

1 Documentation des langues en danger

Nous allons dans cette partie expliciter la notion de « langue en danger » afin de comprendre ce que recouvre précisément ce terme. Nous ferons par la suite la distinction entre « description d'une langue » d'une part et « documentation d'une langue » d'autre part. Bien que ces termes semblent proches, ils recouvrent des réalités différentes qu'il est nécessaire de distinguer. Nous comparerons pour finir la notion de « langue en danger » à celle de « langue peu dotée » pour comprendre ce qui distingue, mais également ce qui rapproche ces deux notions.

1.1 Définitions

1.1.1 Langue en danger

Selon l'UNESCO, une langue peut être considérée comme en danger lorsque « ses locuteurs cessent de l'utiliser, l'utilisent dans de moins en moins de domaines, utilisent de moins en moins ses différents registres et styles de parole, et/ou arrêtent de la transmettre à la génération suivante » (UNESCO, n.d.).

Comme le précise L. Grenoble dans l'ouvrage d'Austin et Sallabank, il ne faut pas avoir une vision manichéenne consistant à dire qu'une langue est soit complètement en danger, soit complètement hors de danger (Austin et al., 2011). En effet, il serait plus juste d'apprécier le niveau de vitalité d'une langue sur un continuum. Ce continuum – mis au point par l'UNESCO – serait le suivant :

- « Sûre ;
- Précaire ;
- En danger ;
- Sérieusement en danger ;
- Moribonde ;
- Morte » (Brenzinger et al., 2003).

Bradley précise que « la terminologie sur les degrés de danger est très diverse, et souvent contradictoire » (Austin et al., 2011). Ainsi, il n'existe pas une seule échelle permettant d'évaluer le niveau de vitalité d'une langue. Toutefois, selon l'auteur, l'échelle de l'UNESCO est le « standard actuel » (Austin et al., 2011).

Un groupe d'experts de l'UNESCO a défini neuf critères permettant d'évaluer le degré de vitalité d'une langue. Chacun de ces critères se voit attribuer une note allant de zéro à cinq – zéro étant le minimum et cinq le maximum. Plus la note est haute, plus le facteur participe à la pérennité de langue :

- « Transmission de la langue d'une génération à l'autre ;
- Nombre absolu de locuteurs ;
- Taux de locuteurs sur l'ensemble de la population ;
- Utilisation de la langue dans les différents domaines publics et privés ;
- Réaction face aux nouveaux domaines et médias ;
- Matériels d'apprentissage et d'enseignement des langues ;
- Attitudes et politiques linguistiques au niveau du gouvernement et des institutions – usage et statut officiels ;
- Attitude des membres de la communauté vis-à-vis de leur propre langue ;
- Type et qualité de la documentation. » (Brenzinger et al., 2003)

Selon Austin et Sallabank, le critère le plus déterminant pour estimer le niveau de vitalité d'une langue reste celui de la transmission intergénérationnelle : « ce facteur est généralement accepté comme étant le *gold standard* de la vitalité d'une langue » (Fishman 1991 cité par Austin et al., 2011). Cela rentre toutefois en contradiction avec les recommandations de l'UNESCO qui précise qu'aucun des critères « ne doit être pris séparément » (Brenzinger et al., 2003). En effet, ils précisent qu' « une langue haut placée selon un certain critère peut réclamer d'urgence une attention immédiate pour d'autres raisons » (Brenzinger et al., 2003). Toutefois, même s'il convient de prendre les critères comme un tout, l'UNESCO précise également que les six premiers sont les plus importants lorsqu'il s'agit d'évaluer le degré de vitalité d'une langue. Plus que le degré de vitalité d'une langue, ils permettent également de prendre en compte les freins à lever ainsi que le travail qui sera à faire en priorité pour sauvegarder la langue.

1.1.1.1 Types de disparition

(Austin et Sallabank, 2011) précisent que la disparition d'une langue est un phénomène qui peut être plus ou moins rapide. (Tsunoda, 2006) distingue ainsi quatre types de disparitions possibles :

- Mort subite : une langue disparaît à cause de la disparition subite de tous ses locuteurs ;
- Mort graduelle : contrairement à la mort subite, la population disparaît de manière graduelle, jusqu'à ce qu'il ne subsiste plus aucun locuteur de la langue ;
- Conversion linguistique subite : les locuteurs d'une langue se mettent subitement à utiliser une autre langue ;
- Conversion linguistique graduelle : les locuteurs d'une langue se mettent graduellement à utiliser une autre langue « suite à un contact linguistique » (Tsunoda, 2006). (Austin et al., 2011) précisent que lorsqu'il y a une conversion linguistique, c'est toujours au déficit des langues parlées par des minorités. En effet, la conversion linguistique se fait toujours au profit d'une langue parlée par un nombre important de locuteurs :

  > *« Les pouvoirs économiques, politiques, sociaux et culturels ont tendance à être dans les mains des locuteurs des langues majoritaires, alors que les locuteurs des langues minoritaires sont marginalisés et leurs locuteurs sous pressions afin qu'ils changent de langue et parlent une des langues dominantes* » Austin et Sallabank (Austin et al., 2011)

Il est également possible de distinguer deux modalités de disparition dans la conversion linguistique:

- « top-down » : la langue n'est d'abord plus parlée par l'élite, et cela se répercute dans toutes les couches inférieures de la société ;
- « bottom-up » : la langue cesse d'abord d'être utilisée dans la sphère privée, avant de ne plus subsister que dans les hautes sphères.

Tous ces types de disparitions ont différents effets pour la langue en question. (Sands et al., 2007) précisent par exemple qu'en cas de mort subite, la disparition de la langue est si rapide que le vocabulaire reste pratiquement inchangé . Dans le cas d'une transition linguistique graduelle, la langue aura une large part de son vocabulaire qui sera empruntée à la langue de transition. En cas de transition linguistique « bottom-up », à terme, seuls les mots les plus spécialisés resteront : mots utilisés dans des contextes juridiques, éducationnels ou liturgiques selon (Sands et al., 2007), les mots de la vie courante ayant disparu en premier.

1.1.1.2 Causes

Les causes de la disparition des langues sont multiples mais ont un impact sur la rapidité de la disparition. (Austin et al., 2011) identifient quatre causes majeures :

- « Catastrophes naturelles ;
- Guerres et génocides ;
- Répression ouverte et assimilation forcée ;
- Dominance culturelle, politique et/ou économique » (Austin et al., 2011).

Les deux premières auront tendance à entraîner une disparition relativement rapide, même si les disparitions rapides restent relativement rares selon (Austin et al., 2011).

Comme le précisent (Besacier et al., 2014), « même une langue avec 100.000 locuteurs n'est à l'abris de l'extinction ». Ainsi, la disparition d'une langue ne concerne pas seulement les langues qui ont un très petit nombre de locuteurs.

1.1.1.3 Conclusion

Comme on peut le constater, les causes de la disparition d'une langue, et par conséquent son type de disparition ont un impact direct sur différents aspects linguistiques de celle-ci. De plus, le type de disparition aura également un impact sur la qualité des données récoltées ainsi que sur la motivation des locuteurs à sauvegarder leur propre langue.

1.1.2 Documentation et description

Bien que « documentation » et « description » semblent désigner la même chose, ces termes recouvrent des réalités différentes. Il s'agit de bien différencier les deux approches.

(Akinlabi and Connell, 2008) définissent la documentation des langues comme étant « la collecte, l'organisation, la transcription et la traduction de données primaires ». Comme le précise Lehmann, le but de la documentation est que « les données récoltées soient représentatives des structures linguistiques de la langue et donne une idée générale de comment et pour quelles raisons la langue est utilisée. Il s'agit d'avoir une représentation de la langue pour ceux qui n'y auraient pas directement accès » (Lehmann, 1999). La documentation permet donc d'avoir une vision globale de la langue et ce dans un maximum de contextes possibles. Cette étape est d'autant plus importante dans le cas des langues en danger que les locuteurs de celles-ci risquent de disparaître à tout moment, et leur langue devenir inaccessible à jamais.

La description d'une langue consiste quant à elle à « formuler, de la manière la plus générale possible, les structures sous-jacentes aux données linguistiques » (Lehmann, 1999). Les grammaires et les dictionnaires sont le résultat de ce processus de description.

Himmelmann précise que même si les notions de description et de documentation sont différentes, elles ne sont pas forcément « séparables dans la pratique » (Himmelmann, 2012) : la transcription d'une langue implique forcément qu'elle soit un minimum décrite – d'un point de vue phonétique et/ou orthographique – et la description d'une langue ne peut se faire sans un minimum de données issues de la documentation.

1.1.3 Langues peu dotées

On désigne par « langues peu dotées » les langues dont le niveau d'informatisation est très bas. (Berment, 2004) a proposé une méthodologie afin d'établir le niveau d'informatisation d'une langue. Il présente à des locuteurs d'une langue une liste de ressources et leur demande de donner un niveau de criticité et une note à chacune :

- Existence de dictionnaires électroniques ;
- Existence de logiciels de traitement de texte ;
- Existence de logiciels de traitement de l'oral ;
- Existence de logiciel de traduction automatique ;
- Existence de logiciel de reconnaissance optique des caractères.

La moyenne pondérée des notes, nommée indice-σ par Berment, permet de mesurer le niveau d'informatisation d'une langue. Si l'indice-σ est inférieur à 10, la langue est alors considérée comme « peu dotée ». (Berment, 2004) précise toutefois que « l'existence d'indices quantitatifs ne doit pas faire oublier que les frontières entre langues peu dotées, moyennement dotées et bien dotées restent imprécises »

Lê Việt Bắc précise dans sa thèse qu'une langue peu dotée peut également être définie comme une langue qui ne « possède pas encore ou pas beaucoup – en quantité et en qualité – de ressources linguistiques » (Le, 2006). Il précise que ces langues sont en général peu dotées informatiquement, c'est-à-dire que leur indice-σ est inférieur à 10.

T. Pellegrini corrèle également le terme de « langues peu dotées » à la « facilité de collecter des données » (Pellegrini, 2008). Ainsi, la représentation d'une langue sur le Web est également un facteur à prendre en compte pour définir la notion de langue peu dotée. . (Pellegrini, 2008) précise que « le fait qu'une langue soit déclarée langue officielle, ou qu'elle soit parlée par un grand nombre de locuteurs, n'implique pas forcément une présence importante de cette langue dans les médias et

en particulier sur Internet. ». C'est donc bien la disponibilité des ressources qui fait qu'une langue peut être considérée comme bien dotée ou non et non son statut.

1.1.4 Conclusion

La notion de langue peu dotée est donc différente de celle de langue en danger. Une langue peu dotée n'est pas forcément en danger. D'ailleurs, nombreuses sont les langues peu dotées avec un niveau de vitalité très élevé (citons notamment le vietnamien, le khmer ou bien encore l'hindi).

Il convient également de distinguer la notion de langue en danger de celle de langue peu documentée. Même si beaucoup de langues sur le point de disparaître actuellement sont peu documentées, il existe tout de même des langues en danger qui sont documentées. A ce titre, le breton en est un bon exemple. Il est en effet classé comme « gravement en danger » par l'UNESCO mais dispose de grammaires, de dictionnaires, de méthodes de langues, etc. De même, cette langue n'est pas forcément peu dotée, puisqu'il existe des dictionnaires électroniques, des traducteurs automatiques et des systèmes de correction orthographique.

Nous pouvons résumer le lien qui unit langues en danger, langues peu documentées et langues peu dotées par le schéma suivant :

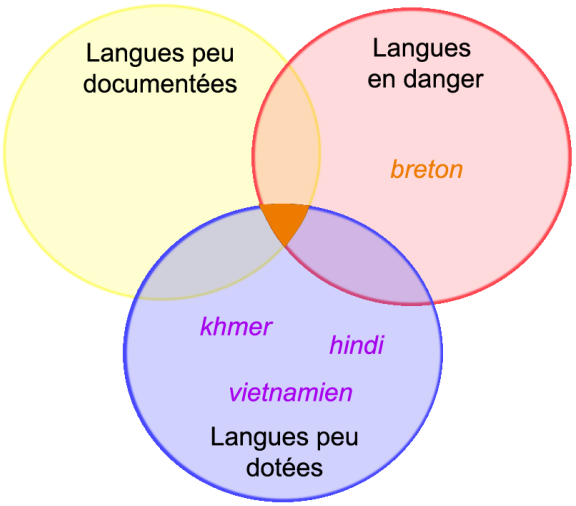


Figure 2 Rapports entre les langues en danger, les langues peu dotées et les langues peu documentées

La pastille orange au centre représente le cas extrême où la langue que l'on étudie est à la fois en danger, peu documentée et peu dotée. Pour ces cas, il est nécessaire de trouver des traitements informatiques qui puissent travailler avec le minimum de ressources possibles comme nous le verrons par la suite.

1.2 Approche classique pour la documentation et la description des langues en danger

En règle générale, la documentation implique qu'un linguiste se déplace dans la communauté de locuteurs dont il souhaite documenter la langue. On dit alors que le linguiste fait du « travail de terrain ».

La documentation d'une langue est un processus complexe qui doit être préparé. Selon Claire Bowern, la documentation « va bien au-delà de la simple récolte de données » (Bowern, 2008). Le travail de documentation doit être préparé bien en amont. Il faut savoir quelles données on compte récupérer, pourquoi et comment on va le faire. Il est également nécessaire selon elle « d'observer les interactions humaines et les pratiques culturelles » (Bowern, 2008).

En général, le schéma type de documentation d'une langue, passe par les activités suivantes :

- « Enregistrement : de médias et de textes (en incluant des méta-données) en contexte ;
- Transfert : vers un espace de gestion des données ;
- Enrichissement: transcription, traduction et annotation des données ;
- Archivage des données ;
- Mobilisation : création, publication et distribution de fichiers […] pour différents utilisateurs et différents usages » (Austin, 2010).

Selon Mosel la première tâche pour récolter des données passe par l'élicitation. L'élicitation consiste « à obtenir des données linguistiques de locuteurs natifs en posant des questions » (Gippert et al., 2006). Toutefois, l'élicitation peut avoir un spectre plus large et peut également être produite en montrant des images ou des vidéos, en faisant écouter de la parole ou bien encore en montrant du texte. L'élicitation consiste donc à inciter un locuteur à produire de la parole.

Selon Mosel, il faut d'abord constituer une liste de mots basiques puis faire en sorte que « les locuteurs natifs produisent de courtes phrases à partir de la liste de mots » (Gippert et al., 2006). Ces courtes phrases permettront notamment d'amorcer l'étude de la syntaxe et de la morphologie. L'auteur préconise également d'enregistrer des contes et histoires puisque la grammaire et le vocabulaire sont généralement simple dans ce type d'écrits.

Mosel envisage la documentation et la description d'une langue par le lien « professeur-élève ». Le locuteur enseigne sa langue au linguiste. Le linguiste en faisant des erreurs lors de son apprentissage permet également au locuteur de prendre conscience des structures de sa propre langue.

Récolter des données de qualité et en quantité est important pour la description des langues. Selon (Bird et al., 2014) « un corpus de textes conséquent peut être à la base de la préparation de grammaires et de dictionnaires, y compris lorsque la langue est éteinte ». Ainsi, même une fois la langue disparue, il sera toujours possible de l'étudier. C'est la raison pour laquelle la documentation doit être de qualité.

1.3 La machine pour assister le linguiste de terrain

Nous allons maintenant nous intéresser aux méthodes faisant appel aux nouvelles technologies pour la documentation et la description des langues en danger.

1.3.1 Récolte des données

« L'augmentation exponentielle de l'usage des smartphones et des tablettes » (Drude et al., 2013) a permis d'appliquer des méthodes de *crowd-sourcing* pour documenter des langues en danger. Il s'agit donc d'impliquer les locuteurs dans la préservation de leur propre langue en leur donnant un outil permettant de le faire.

Une bonne application de *crowd-sourcing* devrait notamment permettre selon l'auteur (Drude et al., 2013) de :

- Créer des données (notamment par élicitation) et de les stocker : le linguiste de terrain n'a plus forcément besoin d'être présent pour la récolte de données ;
- De les traiter : cela inclut notamment l'annotation et la traduction des données. Plus il y a de personnes impliquées, plus le traitement peut être fait rapidement. En effet, le traitement manuel des données audio (transcription phonétique, transcription orthographique si la langue dispose d'un système d'écriture, annotation, etc.) requiert « plus de 100 fois le temps de la durée de l'enregistrement » ;
- De les rendre publiques.

C'est dans ce contexte que l'auteur a créé l'application *Ma Iwaidja* qui permet d'enregistrer des phrases en Iwaidja[1], de les transcrire et de les traduire. Elle permet également de transférer les données dans une base de données afin qu'elles puissent être validées et traduites ou annotées si besoin. (Birch, 2013)

(Bird et al., 2014) ont créé une application, *Aikuma*, permettant également de récolter des données. Contrairement à la précédente, celle-ci n'est pas destinée à une langue en particulier, mais peut au contraire être utilisée par les locuteurs de n'importe quelle langue. Cette application permet d'enregistrer des locuteurs, puis de transcrire et de traduire les enregistrements. Les métadonnées (informations sur le locuteur) sont également attachées à l'enregistrement. Les enregistrements et annotations peuvent être transférés d'un téléphone à l'autre afin que tous les membres de la communauté puissent, s'ils le souhaitent, participer à l'annotation et la traduction des enregistrements.

Cette application propose une fonctionnalité supplémentaire : le *respeaking*. C'est une opération qui consiste à écouter un enregistrement puis à réenregistrer ce qui a été dit dans un environnement calme. En effet, la plupart du temps les enregistrements sont effectués « en contexte, sur des appareils de moindre qualité avec des bruits de fond » (Bird et al., 2014). L'étape de *respeaking* permet d'avoir « un enregistrement audio plus clair, qui facilite le travail de transcription » (Bird et al., 2014).

L'application *LIG-Aikuma*, développée par (Blachon et al., 2016) ajoute des fonctionnalités à la version développée par (Bird et al., 2014). La principale différence entre ces deux applications, outre une interface plus ergonomique, est l'ajout de l'option d'élicitation à partir d'un texte, d'une image, ou d'une vidéo. Cette fonction a été utilisée avec succès lors d'une récolte de données en Mboshi[2] (Blachon et al., 2016).

1.3.2 Traitement des données

Comme mentionné dans (Adda et al., 2016), il est possible de constituer l'inventaire phonologique d'une langue à partir de modèles multilingues. Ces modèles ont été entraînés sur de nombreuses langues et contiennent un large panel des phonèmes existants. Cela permet donc de repérer dans une langue à décrire des phonèmes qu'elle partage avec les langues présentes dans le modèle

[1] Langue en danger parlée en Australie

[2] Langue en danger parlée en République du Congo

multilingue. Ce modèle a toutefois des limites, si la langue à décrire présente un phonème jamais vu auparavant, il ne sera pas possible de le repérer correctement.

Nous aborderons dans la partie suivante les traitements informatiques qui peuvent être appliqués afin de découvrir du lexique de manière automatique.

2 Découverte non supervisée de lexique

2.1 Définitions

Nous allons dans cette partie nous attacher à définir précisément les différents éléments de notre sujet, à savoir la « découverte non supervisée de lexique ». Nous allons tout d'abord définir la notion de « lexique » et également voir ce qui distingue cette notion de celle de « vocabulaire ». Nous définirons par la suite le terme de « découverte non supervisée ». Nous terminerons en introduisant la notion de « *zero resource* » et verrons en quoi cette notion est fondamentale dans le cadre de notre approche.

2.1.1 Lexique et vocabulaire

Le terme « lexique » est défini comme étant « une entité théorique correspondant à l'ensemble des lexies de cette langue » (Polguère, 2001). Les lexies désignent ici les unités lexicales d'une langue, également appelées « lemmes ». Un certain nombre de « mots formes » peut être associé à chaque lemme. Les mots formes résultent de l'association à un lemme d'un ou plusieurs morphèmes de flexion. Nous désignerons par la suite par « forme fléchie » la notion de « mot forme ».

Il est nécessaire de distinguer la notion de « lexique » de celle de « vocabulaire ». Polguère distingue « vocabulaire d'un texte » et « vocabulaire d'un individu », nous ne nous intéresserons ici qu'à ce dernier. Le « vocabulaire d'un individu » se définit comme « le sous-ensemble du lexique d'une langue donnée contenant les lexies de cette langue que maîtrise l'individu en question » (Polguère, 2001).

Ainsi, un medium – qu'il soit textuel ou oral – nous permet d'accéder à diverses formes fléchies utilisées par un individu. Ces formes fléchies dénotent la capacité d'un individu à utiliser un certain nombre de lemmes. En compilant ces lemmes, nous constituons le vocabulaire d'un individu. L'étude d'un certain nombre de personnes permet, par la compilation du vocabulaire de celles-ci, de constituer le lexique de la langue qu'elles parlent.

2.1.2 Découverte non supervisée

L'apprentissage non supervisé désigne une méthode d'apprentissage automatique. Elle s'oppose à l'apprentissage supervisé et à l'apprentissage semi-supervisé. Dans le cadre d'un apprentissage non supervisé, l'on dispose d'un ensemble de données qui doivent être regroupées en différentes classes. Il s'agit alors de repérer des instances de type similaire afin de les regrouper. Cette opération est appelée *clustering*.

Dans notre cas, il s'agit donc d'extraire automatiquement, à partir des données d'entrée (quel que soit leur type), un ou plusieurs mots de la langue étudiée : chaque classe correspond ici à un mot.

2.1.3 Approche « *zero resource* »

L'emploi du terme « *zero resource* » est assez récent et vient de l'organisation de la compétition « The Zero Resource Speech Challenge ».

Comme son nom l'indique, il s'agit de n'utiliser strictement aucune ressource habituellement utilisée dans le domaine du traitement du signal vocal. En effet, les systèmes de reconnaissance vocale s'appuient généralement sur trois ressources principales : un modèle de langue, un modèle acoustique et un modèle de prononciation. Ainsi, les systèmes participant à la compétition cherchent à se passer de ces trois modèles. Le site internet de la compétition précise qu'il s'agit « d'utiliser uniquement les informations dont dispose un enfant apprenant une langue [maternelle] »[3]. Les organisateurs précisent également que « zero resource » signifie « aucune connaissance linguistique préalable » et que l'on peut librement se servir d'autres sources d'informations, comme la vision par exemple.

L'utilisation de modèles adoptant une telle approche serait nécessaire dans le cadre de nos recherches. Comme nous l'avons vu précédemment, les langues en danger sont généralement peu documentées et donc les ressources dont nous aurions besoin pour constituer les modèles habituels sont donc inexistantes.

2.2 Segmentation d'un signal d'entrée en sous-unités

Notre travail de recherche portera sur les langues en danger. Comme précisé auparavant, peu d'entre elles disposent d'un système d'écriture ou de ressources qui pourraient nous être utiles, tels que des dictionnaires ou des corpus. Ainsi, nous nous intéresserons aux travaux ayant une approche *zero resource*.

[3] http://sapience.dec.ens.fr/bootphon/ [consulté le 24 mai 2017]

2.2.1 Approches

(Lee et al., 2016) distinguent deux approches dans la découverte non supervisée de mots : *spoken term discovery* et *word segmentation*.

La première, *spoken term discovery*, consiste à identifier dans un signal vocal des motifs qui se répètent et par conséquent à « trouver des mots clefs dans le signal qui étaient précédemment inconnus » (Lee et al., 2016). Les mots clefs ne correspondent pas forcément à des mots, c'est-à-dire à une forme fléchie unique, mais peuvent être une suite de plusieurs mots récurrents (collocation). La méthode classique consiste « à identifier les sous unités de l'espace acoustique qui sont similaires, puis à les clustériser pour découvrir des catégories qui correspondent à des unités lexicales » (Lee et al., 2016). Les modèles adoptant une telle approche peuvent donc travailler directement sur le signal sans qu'il ait été transcrit au préalable.

La seconde, *word segmentation*, consiste à « partir d'une chaine de symboles non segmentée et à essayer d'identifier des sous unités correspondant à des unités lexicales » (Lee et al., 2016). Ces modèles ne peuvent donc travailler que sur une chaine de symboles et non sur du signal brut. Cette chaine de symboles peut aussi bien être les graphèmes d'un système d'écriture quelconque ou une suite de symboles phonétique. Ainsi, pour pouvoir segmenter un signal d'entrée en sous unités, il est nécessaire de le transcrire au moyen d'un système de décodage acoustco-phonétique (DAP).

Les auteurs considèrent qu'il existe une troisième approche : *unsupervised lexicon discovery*. En réalité, il convient plus de la considérer comme une extension de la précédente. A la différence des deux approches précédentes, celle-ci permet de travailler directement sur le signal (à la différence de l'approche *word segmentation*) et permet d'apprendre différents niveaux de structures linguistiques (à la différence de *spoken term discovery*).

2.2.2 Quelles difficultés ?

(Abdellah Fourtassi et al., 2013) ont remarqué que les résultats d'une segmentation effectuée automatiquement peut varier d'une langue à l'autre, et ce pour une même méthode de segmentation. Ils ont en effet constaté que la segmentation du japonais était considérablement moins bonne que l'anglais. Ils expliquent cette différence par la structure syllabique des langues en question. L'anglais autorise un certain nombre de consonnes à se suivre, là où le japonais n'autorise que des groupes du type CV. De ce fait le japonais « doit utiliser des mots multi syllabiques pour avoir un lexique d'une taille conséquente, là où l'anglais pourrait, en principe, utiliser des mots monosyllabiques ». Les mots étant plus longs, il y a plus de chances que ceux-ci incluent dans leur forme des mots plus petits. Cela entrainerait par conséquent une hyper-segmentation.

Ainsi, la structure syllabique de la langue a un impact fort sur la formation du lexique. Cette donnée est à prendre en compte pour avoir une segmentation correcte et ainsi éviter de trop nombreuses hypo- ou hyper-segmentations. Il conviendra donc de sélectionner les méthodes de segmentation selon la langue.

2.3 Méthodes générales

2.3.1 Information mutuelle

L'utilisation de l'information mutuelle pour la segmentation en mots a été utilisée par (Besacier et al., 2006). Ce modèle repose sur la prédictibilité des phonèmes : « Le nombre de phonèmes distincts qui sont les possibles successeurs d'une chaîne décroît rapidement à mesure que la chaîne grandit, sauf si une frontière de morphème a été franchie » (Besacier et al., 2006). Cette propriété linguistique peut être formalisée mathématiquement en utilisant l'information mutuelle qui « est une quantité mesurant la dépendance statistique de [deux] variables [aléatoires] »[4]. Le système tient également compte des frontières de mots déjà présentes, à savoir les frontières de début de mot et de fin de mot situées respectivement au début et à la fin de chaque phrase.

Le modèle développé ne fonctionne pas sur du signal brut, il est nécessaire que le signal soit auparavant transcrit en symbole phonétiques. Cette approche n'est donc pas *zero resource* puisqu'elle nécessite l'utilisation d'un système *speech to text*, qui a besoin d'un modèle de langue, d'un modèle de prononciation et d'un modèle acoustique pour fonctionner.

2.3.2 Regroupement de syllabes

Nous allons maintenant nous intéresser à un modèle qui travaille directement sur du signal de parole développé par (Räsänen et al., 2015). Ce modèle permet de faire découvrir des mots de manière non supervisée et ne requiert aucune information linguistique sur la langue étudiée. Cette approche se base sur le regroupement de syllabes.

La première étape de leur modèle consiste à segmenter le signal d'entrée en une suite de syllabes. Pour ce faire ils calculent l'enveloppe de l'amplitude du signal. Une fenêtre glissante permet de repérer les minimas locaux qui deviennent des frontières de syllabes. Les MFCC (Mel-Frequency Cepstral Coefficients) de chacune des syllabes sont ensuite calculés afin de pourvoir les regrouper. La dernière étape de leur modèle consiste à retrouver les unités lexicales en utilisant des n-grammes pour « trouver des groupes de syllabes récurrents » (Räsänen et al., 2015).

[4] https://fr.wikipedia.org/wiki/Information_mutuelle [consulté le 24 mai 2017]

Ce modèle a été testé sur deux langues, l'anglais et le tsonga. Ce dernier présente des caractéristiques similaires au japonais qui a un « plus haut taux de mots multi syllabiques » (Räsänen et al., 2015) que l'anglais. Ce modèle permet de prédire des frontières de mots avec une F-mesure de 46.7 % et 33.5 % pour l'anglais et le tsonga respectivement. Cela montre donc que « l'enveloppe de l'amplitude d'un signal contient de forts indices pour la segmentation en mots dans un contexte purement non-supervisé, permettant le découpage en syllabes et par conséquent la découverte de frontières de mots » (Räsänen et al., 2015). Toutefois, même si un certain nombre de frontières sont correctement posées et certains mots devinés, leur approche ne permet pas de couvrir l'ensemble du corpus de test et ainsi une large partie reste non segmentée.

2.3.3 Approches par réseaux de neurones

La segmentation d'une chaîne en sous unités peut également être faite par des réseaux de neurones. Les réseaux de neurones visent à reproduire numériquement le fonctionnement d'un réseau de neurones biologiques. Les réseaux de neurones sont constitués de plusieurs neurones organisés en couches. Chaque neurone donne une valeur de sortie qui est calculée en fonction des valeurs d'entrées qui ont préalablement été pondérées par un certain poids. Afin de pouvoir fonctionner, les réseaux de neurones doivent d'abord être entraînés sur un corpus d'apprentissage, où ils apprendront quelles sorties prédire en fonction des entrées.

L'approche employée par (Gelderloss and Chrupala, 2016) utilise un GRU (Gated Recurrent Unit) à trois couches. Le but de leur modèle est « de prédire les caractéristiques visuelles d'une image à partir de sa description sous forme d'une séquence de phonèmes »(Gelderloss and Chrupala, 2016)

Les GRU « calculent la valeur d'un état caché à un moment h_t par la combinaison linéaire de l'activation précédente h_{t-1} et du nouveau candidat à l'activation $\tilde{h}_t$ » (Gelderloss and Chrupala, 2016). Cela permet donc de donner une mémoire au réseau de neurones et lui permet de prendre des décisions en fonction de l'entrée en cours et de l'entrée précédente.

Le GRU ne permet pas de modéliser directement les frontières de mots. C'est la raison pour laquelle il a fallu utiliser un système de régression logistique qui a appris où étaient les frontières en se servant des valeurs de sortie de chacune des couches du GRU. Le système a été entraîné sur un corpus d'apprentissage où étaient indiquées les frontières de mots.

Ce système permet de prédire des frontières de mots de manière correcte avec une précision de 88 %. Les informations linguistiques encodées présentes dans la première couche du réseau de neurones permettent de prédire les frontières de mots avec la même précision qu'un modèle n-gramme d'ordre 1.

L'approche de (Gelderloss and Chrupala, 2016) nécessite toutefois d'avoir la représentation phonétique du signal en entrée et ne permet pas de travailler directement sur le signal brut. De plus, comme pour toute approche par réseaux de neurones, il faut disposer d'un corpus suffisamment représentatif pour l'apprentissage.

On peut également citer l'approche employée par (Duong et al., 2016) et (Bérard et al., 2016) qui consiste à employer un réseau de neurones LSTM (*Long Short-Term Memory*) bidirectionnels. Les LSTM ont pour particularité de mémoriser certaines informations pour une durée de temps variable. Cela permet de prendre des décisions à moment *t* à partir des informations survenues à un moment *t-1*, *t-2*, etc.

Ces deux travaux ont pour but de faire de la traduction automatique avec en entrée un signal vocal ou un texte dans une langue source et d'obtenir en sortie sa traduction dans une langue cible. Lorsque ces modèles se servent d'un signal en entrée, ils ne requièrent pas que celui-ci soit transcrit. La partie qui nous intéresse dans ces modèles est la partie d'alignement qui est faite lors de l'apprentissage, qui permet d'apprendre à faire une segmentation.

Ces modèles sont constitués de trois parties : un encodeur, un modèle d'attention et un décodeur. Le modèle d'attention est la base de ce modèle. Il « encode la source comme une séquence de vecteur, puis fait un décodage pour générer les sorties » (Duong et al., 2016). Le modèle d'attention impose le nombre de sorties qui doivent être générées par l'encodeur. Ce nombre de sorties est égal au nombre de mots dans la langue cible. Le signal d'entrée est transformé en suite de vecteurs (PLP chez (Duong et al., 2016) et MFCC chez (Bérard et al., 2016)). L'encodeur va venir concaténer un certain nombre de vecteurs entre eux afin de produire le nombre de mots désirés en sortie. Ainsi, cela permet implicitement de faire une segmentation du signal d'entrée en mots.

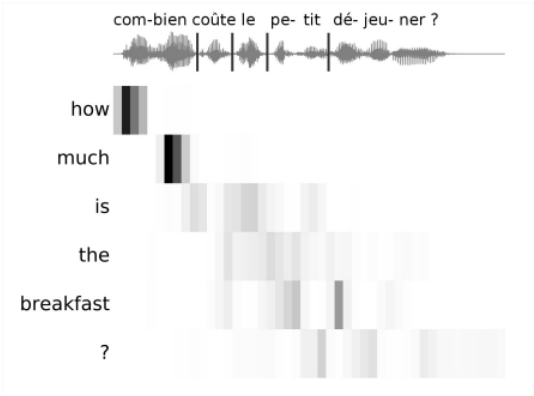


Figure 3 Alignement produit par le modèle de traduction encodeur/décodeur couplé à un modèle d'attention (Bérard et al., 2016)

Dans l'exemple ci-dessus, on constate que le signal d'entrée a été segmenté en différentes unités afin de pouvoir produire une traduction. Toutefois, on peut remarquer que la segmentation n'est pas encore très performante. « Combien » a par exemple été hypersegmenté en « com-bien » afin de pouvoir être aligné avec « how much ». Le reste de la segmentation reste cependant assez flou, comme on peut le voir avec le mot « the » qui semble être aligné avec plusieurs des mots du corpus.

2.3.4 Approches à base d'alignement dynamique (DTW)

Il est également possible de découvrir du lexique grâce au Dynamic Time Warping (DTW). Le DTW se définit comme étant une mesure capable de « mesurer la similarité de deux énoncés directement au niveau acoustique » (Park and Glass, 2005). Le DTW s'applique directement au niveau du signal et non à sa transcription. De plus, un signal seul n'est pas suffisant, il est nécessaire de disposer de deux signaux partageant des mots en commun. Le but est de repérer et d'aligner dans chacun des deux signaux les mots partagés. Cette technique permet donc de faire du *spoken term discovery*, qui s'apparente à de la recherche de mot clé.

Le DTW permet de faire un alignement entre deux énoncés et donne une mesure de similarité qui permet de savoir si les deux énoncés sont proches ou non. Toutefois, l'utilisation brute de la DTW n'est pas très utile pour la découverte non supervisée de lexique puisqu'il considère les énoncés dans leur globalité sans prendre en compte le fait qu'ils peuvent être constitués de plusieurs mots (Park and Glass, 2005). L'alignement n'est donc pas fait mot à mot : « l'algorithme trouve seulement l'alignement global optimal qui unit les deux énoncés, et non les alignements locaux qui correspondent à des sous unités correspondantes » (Park and Glass, 2005). L'utilisation du DTW segmental permet de pallier ce problème, en permettant un alignement, non plus de manière globale, mais de manière locale.

Pour procéder à un tel alignement, il est nécessaire de calculer la matrice de distance DTW d'une manière globale tout d'abord. La matrice de distance est ensuite découpée en bandes diagonales de largeur fixe pour permettre que « les deux sous unités ne soient pas trop éloignées d'un point de vue temporel pendant l'alignement » (Park and Glass, 2005). Il suffit ensuite de calculer la distance DTW de manière locale dans chacune des bandes diagonales et de sélectionner le meilleur alignement. Pour plus d'efficacité, le signal d'entrée est d'abord prétraité afin d'éliminer les portions ne correspondant pas à de la parole (i. e. des silences).

Lorsque deux signaux vocaux présentent suffisamment de similarité (i. e. ils ont au moins un mot en commun), ils sont liés ensemble dans un graphe. Chaque nœud de ce graphe peut être vu comme un nœud qui est le père d'autres nœuds. Ces fils représentent chacun une portion de signal ayant une

similarité avec une portion de signal appartenant à un ou plusieurs autres nœuds pères. Ainsi, les fils de nœuds pères différents sont reliés par un arc qui représente la mesure de similarité obtenue par DTW.

Cette méthode permet donc d'extraire des segments communs et de les relier. Ces segments peuvent aussi bien être des mots ou des groupes de mots ou encore des unités sous-lexicales.

2.3.5 Amélioration de l'approche d'alignement dynamique segmental

L'application de l'alignement dynamique segmental tel que proposé par (Park and Glass, 2005) requiert un temps de calcul élevé. Plus il y aura de signal à traiter, plus les combinaisons possibles de sous-segments à analyser seront nombreuses.

Les améliorations apportées par (Jansen and Durme, 2011) permettent justement de palier ce défaut : la complexité algorithmique de traitement en temps passe de $O(n^2)$ [5] à $O(n \log n)$ et la complexité en espace passe de $O(n^2)$ à $O(n)$. Afin de parvenir à ce résultat, (Jansen and Durme, 2011) retardent au maximum l'application de l'alignement dynamique segmental et passe par une suite d'approximations afin de générer la matrice de similarité de deux signaux.

2.4 Découverte non supervisée de lexique et acquisition du langage

2.4.1 Adaptor Grammars

Avant d'introduire à proprement parler le modèle des *Adaptor Grammars* il convient d'introduire un certain nombre de notions sur lesquelles il se base. Nous parlerons donc tout d'abord de la loi de Zipf, puis nous expliquerons ce que sont les grammaires hors contexte, et leur extension : les grammaires hors contexte probabilistes.

2.4.1.1 Loi de Zipf

Les mots d'une langue n'ont pas la même fréquence d'apparition. Ainsi, lorsque l'on fait une analyse statistique sur la fréquence de mots dans un texte, on se rend compte que certains mots apparaissent très fréquemment (les mots outils et les auxiliaires notamment) alors que d'autres n'apparaissent que très rarement.

[5] « Quand le paramètre double, le temps d'exécution est multiplié par 4 » (https://www.u-picardie.fr/~furst/docs/4-Complexite.pdf [consulté le 24 mai 2017])

Cette observation, mathématiquement formalisée par le linguiste américain George Zipf, est désormais connue sous le nom de *loi de Zipf*. Ainsi, le mot situé au rang *r* a une fréquence *f(r)* équivalente à :

$$f(r) \propto \frac{1}{r^{\alpha}}$$

où $\alpha \approx 1$ (Piantadosi, 2014 d'après Zipf 1936, 1949).

Cette observation est vraie quelle que soit la langue. Toutes les langues du monde ont donc une distribution « zipfienne » de mots.

2.4.1.2 Grammaires hors contexte

Les grammaires hors contexte (GHC) peuvent être définies comme étant un quadruplet V, Σ, R, S où :

- « V est un ensemble fini de symboles non terminaux ;
- Σ est un ensemble fini de symboles terminaux ;
- R est un ensemble fini de règles R ⊆ V × (V ∪ Σ)$*$;
- $S \in V$ est un axiome de la grammaire (élément initial). »[6]

Ces grammaires permettent de reconnaître des langages dits « hors contexte ». Ainsi, à tout langage hors contexte, il existe une grammaire hors contexte qui peut le reconnaître. Ces grammaires sont utilisées en traitement du langage naturel pour savoir si une phrase est bien formée, et le cas échant, obtenir son arbre syntaxique.

Toutefois, le langage naturel est ambigu et plusieurs arbres de dérivations sont possibles pour une seule et même phrase. C'est la raison pour laquelle il existe une extension des grammaires hors contexte : les grammaires hors contexte probabilistes (GHCP). Celles-ci peuvent être définies comme un quintuplet V, Σ, R, S, et P où V, Σ, R, S ayant la même définition que précédemment et où P est une distribution de probabilités associée à chacune des règles de la grammaire. Les probabilités de chacune des règles ont d'abord été apprises sur un corpus d'entraînement. L'intérêt d'avoir des probabilités à chacune des règles est de pouvoir donner la dérivation la plus probable pour une phrase.

Soit la grammaire suivante avec comme phrase à reconnaitre « *Fish people fish tanks with rods* ».

[6] http://www.labri.fr/perso/anca/Langages/cours/cfl.pdf [consulté le 24 mai 2017]

```
S->NP VP(0.9)
S->VP(0.1)
VP->V NP(0.5)
VP->V(0.1)
VP->V @VP_V(0.3)
VP->V PP(0.1)
@VP_V->NP PP(1.0)
NP->NP NP(0.1)
NP->NP PP(0.2)
NP->N(0.7)
PP->P NP(1.0)

N->people(0.5)
N->fish(0.2)
N->tanks(0.2)
N->rods(0.1)
V->people(0.1)
V->fish(0.6)
V->tanks(0.3)
P->with(1.0)
P->in(1.0)
```

[7]

Figure 4 Exemple de grammaire hors contexte probabiliste

L'analyse de la phrase nous donne comme résultat l'arbre de dérivation le plus probable étant donné les probabilités de chacune des règles.

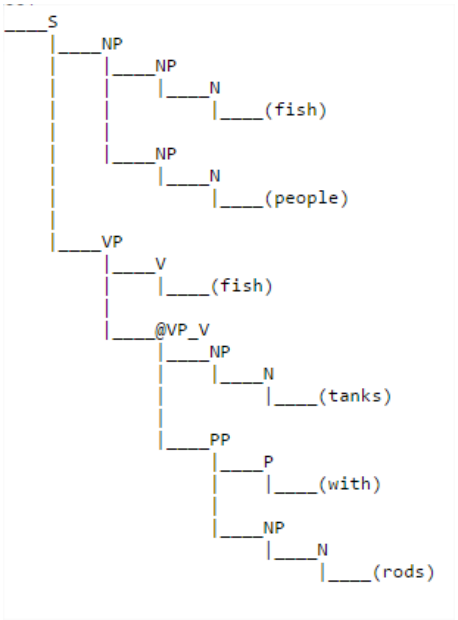


Figure 5 Arbre de dérivation produit par la grammaire précédente pour la phrase "Fish people fish tanks with rods"

[7] D'après la grammaire présentée dans la vidéo de Dan Jurafski : https://www.youtube.com/watch?v=hq80J8kBg-Y [consulté le 24 mai 2017]

2.4.1.3 Modèle générateur/adaptateur

Le modèle générateur/adaptateur a été développé par (Goldwater, 2006). Ce modèle permet de générer des séquences de mots, et permet par conséquent de faire une segmentation à partir d'une chaîne non segmentée. Ce modèle est constitué de deux parties : un générateur et un adaptateur.

> « *Un* générateur *[est] un modèle génératif sous-jacent de mots qui ne sont (en général) pas distribués selon une loi de puissance, et un adaptateur transforme ce flux de mots produit par le générateur en un flux dont les fréquences obéissent à une loi de distribution.* » (Goldwater et al., 2011)

Ainsi, on peut considérer le générateur comme le processus qui va venir segmenter la chaîne, et l'adaptateur comme le processus qui va venir contrôler cette segmentation. En l'occurrence, nous souhaitons que les mots produits suivent la distribution de la loi de Zipf. L'adaptateur peut recourir pour ce faire à plusieurs modèles mathématiques. Par exemple, le processus de Dirichlet ou le processus de Pitman-Yor qui est une généralisation de Dirichlet.

Pour mieux faire comprendre comment marche l'adaptateur, (Goldwater, 2006) utilise l'analogie culinaire du processus du « restaurant chinois » :

> « *C'est un restaurant avec un nombre infini de tables, chacune avec un nombre infini de sièges. Les clients rentrent dans le restaurant un à un, et choisissent une table où s'asseoir. La probabilité de s'asseoir à une table est proportionnelle au nombre de personnes déjà assises à cette table, et la probabilité de choisir une table inoccupée est proportionnelle à un paramètre constant α* » (Goldwater et al., 2011)

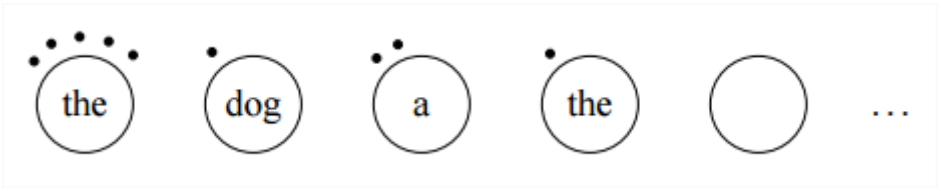


Figure 6 Exemple du processus du "restaurant chinois" appliqué à la segmentation (Goldwater et al., 2011)

Ainsi, dans l'exemple ci-dessus (Goldwater et al., 2011) le générateur/adaptateur a produit pour chacune des tables un mot dont le nombre de points noirs au-dessus est le nombre d'occurrences. Le générateur est chargé de segmenter quand l'adaptateur est lui chargé de créer ou non une nouvelle table. Si une nouvelle table est créée, cela signifie qu'un nouveau mot encore jamais rencontré dans le flux de symboles analysé est apparu. Plus un mot a un nombre d'occurrences élevé, plus il a de chances de réapparaître par la suite selon le principe du « rich-get-richer » (Goldwater et al., 2011)

Ce modèle « en deux phases » comme le nomme Goldwater est le modèle qui a donné naissance aux *Adaptor Grammars*.

2.4.1.4 Lien GHCP et Adaptor Grammar Framework

Les GHCP sont des modèles *paramétriques* puisque le nombre de règles ainsi que les probabilités qui y sont associées sont fixes. Les *Adaptor Grammar*, ou grammaires adaptatives, sont l'équivalent *non paramétrique* des GHCP. Cela signifie que le nombre de règles n'est pas fixe et que les probabilités associées à celles-ci peuvent changer au cours de l'apprentissage.

Les grammaires adaptatives peuvent être définies comme étant un sextuplet V, Σ, R, S, P, α. Les symboles V, Σ, R, S et P sont les mêmes que pour les GHCP. Le paramètre α est le « paramètre de concentration de Dirichlet » (Johnson, 2008a) qui permet de contrôler la création d'une nouvelle règle dans la grammaire et d'imposer une distribution particulière. Le système va naturellement favoriser la réutilisation d'un arbre qui résulte d'une adaptation plutôt que d'en créer un nouveau. Ce paramètre de concentration permet donc de "favoriser la création d'un petit lexique avec plus d'unités lexicales réutilisables » (Lee et al., 2016) .

Comme le précise Johnson « les nouvelles règles apprises par une grammaire adaptative sont des compositions des anciennes règles (qui elles-mêmes peuvent être des compositions d'autres règles), […] on peut donc considérer les nouvelles règles comme étant des fragments d'arbres, ou chaque fragment est associé à une probabilité » (Johnson, 2008a)

2.4.1.5 Exemples de grammaires adaptatives

Nous allons ici présenter les applications qui ont été faites des grammaires adaptatives sur un corpus de parole de personnes s'adressant à des enfants.

- Modèle unigramme

La grammaire ci-dessus permet de faire une segmentation en mots. Cette grammaire a été mise en œuvre sur le corpus Brent[8] et a obtenu un F-mesure de 56%.

$$\text{Words} \rightarrow \text{Word}^{+}$$
$$\underline{\text{Word}} \rightarrow \text{Phon}^{+}$$

Figure 7 Exemple de grammaire adaptative (Johnson, 2009)

[8] Corpus d'interaction entre des enfants et leurs parents (Brent, M., & Cartwright, T. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. Cognition, 61, 93–125.)

Word désigne un symbole non terminal qui peut être adapté. **Phon** est un symbole terminal qui peut prendre pour valeur chacun des signes phonétiques de l'anglais. Après apprentissage, la grammaire devient :

Words →Word Words
Words →Word
Word → Phon^{+}
Word → *y u*
Word → *I n*
Word → *w I T*
Word → *D 6 d O g i*
Word → *I n D 6*
Word → *I n D 6 h Q s*

Figure 8 Evolution de la grammaire adaptative précédente après apprentissage (Johnson, 2009)

On constate que la grammaire a permis de stocker des sous arbres entiers afin de pouvoir par la suite les réutiliser.

Par exemple, pour la transcription phonétique « D6bUk » correspondant à la phrase « the book » en anglais, la grammaire a permis de faire une segmentation correcte :

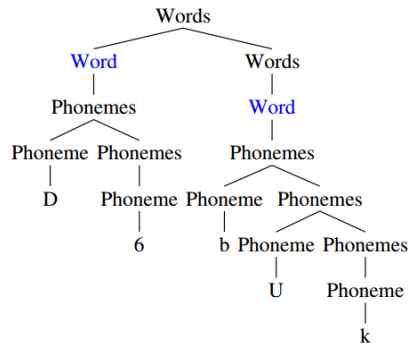


Figure 9 Arbre de dérivation produit par la grammaire adaptative précédente pour la chaîne "D6bUk"

Chaine phonétique «d6 buk» représentant la phrase anglaise « the book »
(Johnson, 2009)

- Modèle bigramme

Le modèle précédent ne prenait pas en compte les mots dans leur contexte et avait tendance à l'hypo-segmentation. En effet, les modèles unigrammes génèrent un mot « en supposant que les mots sont générés indépendamment, i. e. la probabilité de générer un mot en particulier est la même peu importe le mot qui est apparu avant » (Goldwater, 2006). Les chercheurs du domaine ont donc cherché à améliorer la grammaire en tentant de prendre en compte la dépendance entre les mots grâce à des « collocations ». Les collocations permettent de représenter des bi-grammes.

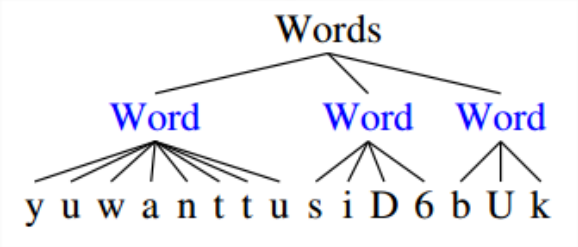


Figure 10 Arbre de dérivation produit par une grammaire adaptative unigramme

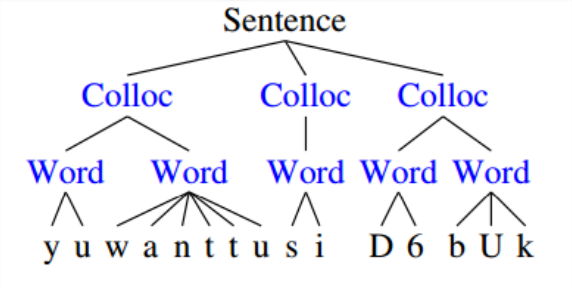


Figure 11 Arbre de dérivation produit par une grammaire adaptative bigramme

Chaîne phonétique « yu want tu si D6 bUk » représentant la phrase « You want to see the book »

(Johnson, 2009)

- Modélisation des syllabes

Ce modèle est suffisamment flexible pour permettre de générer des mots en fonction de leur structure phonologique. Il permet par exemple de représenter les syllabes afin de produire une segmentation correcte.

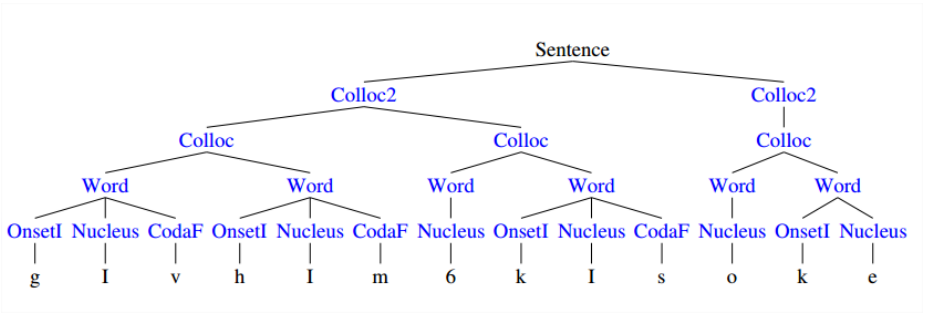


Figure 12 Arbre de dérivation produit par une grammaire adaptative modélisant les syllabes

Chaîne phonétique « gIv hIm 6 kIs oke » représentant la phrase anglaise « Give him a kiss okay »

(Johnson, 2009)

2.4.1.6 Intérêt des grammaires adaptatives

Les grammaires adaptatives permettent de faire des analyses avec des niveaux de granularité plus ou moins fins. Toute comme les GPHC peuvent être utilisées pour la reconnaissance d'un langage (et produire son arbre de dérivation), les grammaires adaptatives peuvent également être utilisées de la même manière. Ainsi, ce modèle est utilisé notamment pour segmenter une chaîne en mots ou pour des analyses morphologiques.

Comme le précise Shay Cohen, l'intérêt principal des grammaires adaptatives est « qu'elles permettent de traiter un flux de données continu, sans avoir besoin de relire plusieurs fois les données, ou de les garder toutes en mémoire. » (Cohen, 2016)

Comme le note (Johnson, 2008b) : « il y a des interactions synergiques en apprenant différents niveaux de structures linguistiques de manière simultanée, par comparaison à l'apprentissage de chacun des différents niveaux de structures linguistiques de manière indépendante ». C'est pour cela que le modèle des grammaires adaptatives est particulièrement performant. Il permet de modéliser différents niveaux de manière dynamique. Toutefois, ces différents niveaux d'apprentissage doivent être définis au préalable.

Les résultats produits par les grammaires adaptatives nous semblent particulièrement cohérents avec les observations faites par (Bannard and Matthews, 2008) qui stipulent que les enfants ont tendances à mémoriser des séquences entières de mots comme un tout. Ainsi, les enfants ne procèderaient pas à une segmentation en mots mais stockeraient en mémoire différents niveaux de segmentations, allant de chunks entiers en passant par des n-grammes fréquents. La segmentation en collocation comme le montre la Figure 12 semblent être particulièrement pertinente pour modéliser l'acquisition du langage.

Cependant, les grammaires adaptatives telles que présentées ci-dessus ne permettent pas de traiter du signal brut. Celui-ci doit d'abord être transcrit en symboles phonétiques. De plus, les grammaires adaptatives requièrent qu'il y ait déjà une segmentation en phrases de faite.

2.4.1.7 Traitements directement sur le signal

(Lee et al., 2016) ont proposé un modèle de découverte non supervisée de lexique à partir du signal en se servant des grammaires adaptatives. Ce modèle est un modèle « zero resource » car il ne se sert ni d'un modèle de prononciation, ni d'un modèle acoustique ni d'un modèle de langue.

Le point principal de leur modèle est qu'il cherche à modéliser les variations de prononciation, notamment dues à la coarticulation et à l'intonation. Il est nécessaire de modéliser cette variabilité « pour reconnaître ces prononciations comme étant les instances d'un même type de mot » (Lee et al., 2016)

La première étape de ce modèle consiste à transformer le signal en suite de « pseudo-phones ». Ceux-ci sont modélisés par leurs coefficients MFCC. Chaque cluster se voit attribuer un identifiant. La seconde étape consiste à modéliser la variabilité des pseudo-phones en jouant sur la longueur de ceux-ci. Il est possible de spécifier au modèle le nombre de pseudo-phones finaux que l'on souhaite

obtenir ou de laisser le modèle décider du nombre de phonèmes finaux en imposant un paramètre de concentration de Dirichlet. Cela permet d' « inférer un inventaire des phonèmes » (Lee et al., 2016) et de « modéliser la réalisation vocale de chaque unité phonétique dans l'espace vectoriel » (Lee et al., 2016).

Une fois ceci fait, la séquence de pseudo-phones est donnée à la grammaire adaptative qui produit une segmentation. Cette segmentation de pseudo-phones permet donc de faire une segmentation du signal.

La segmentation faite par ce modèle obtient un F-mesure de 20 (au mieux) ce qui reste tout de même relativement faible. Les auteurs précisent toutefois que « les unités découvertes par le modèle font sens d'un point de vue linguistique, même si elles ne correspondent pas toujours strictement à des frontières de mots (i.e. les unités peuvent êtres de morphèmes ou des collocations, etc.) » (Lee et al., 2016)

2.4.1.8 Aide des informations prosodiques

(Ludusan et al., 2015) ont montré que les informations prosodiques pouvaient également être des indices permettant d'améliorer la segmentation. Leur modèle permet d'intégrer des informations prosodiques directement dans une grammaire adaptative. De plus ce modèle reste « zero resource » puisque les informations prosodiques sont extraites directement du signal, sans l'aide d'une information linguistique annexe.

La première étape du modèle consiste à poser des frontières prosodiques le long du signal. Pour ce faire, les auteurs se servent de quatre éléments que sont « les pauses silencieuses, l'allongement final, l'allongement initial et le retour de F0 à sa valeur d'origine » (Ludusan et al., 2015) pour détecter des frontières prosodiques. Ces éléments sont calculés, puis normalisés et additionnés pour chacune des syllabes présentes dans le signal. Une fenêtre glissante vient ensuite poser des frontières prosodiques lorsqu'un maximum local est atteint.

Le modèle de grammaire adaptative utilisé est un modèle utilisé par (Johnson and Goldwater, 2009). Cette version a été modifiée afin de pouvoir intégrer la présence d'une frontière prosodique.

Le modèle a été testé sur deux langues différentes : le japonais et l'anglais. Les résultats sont nettement meilleurs pour le japonais que pour l'anglais. Les auteurs expliquent cela par le fait qu' « il est possible que la prosodie aide particulièrement les langues dont le lexique est polysyllabique, en empêchant d'avoir une hyper-segmentation ».

Toutefois, l'inclusion des frontières prosodiques dans la grammaire a un impact différent selon l'endroit où elles sont placées, ce paramètre variant d'une langue à l'autre. Ainsi, si l'on n'a aucune connaissance de la langue que l'on souhaite segmenter, il apparaît difficile de savoir à quel niveau placer les frontières prosodiques dans la grammaire.

2.4.2 Acquisition du langage et corpus multimodaux

Le travail de (Roy and Pentland, 2002) s'inscrit dans un contexte multimodal puisque leur corpus est constitué d'images et de signaux vocaux. Leur étude vise à modéliser l'apprentissage du langage chez les enfants.

Leur méthodologie est la suivante : chaque interaction enfant-adulte est enregistrée et reliée à l'objet avec lequel ils sont en train de jouer. Chaque objet a d'abord été photographié. Cela a servi à avoir une représentation abstraite de l'objet manipulé à fournir au système. L'apprentissage des mots est fait « par un processus de mémoire à deux niveaux, une mémoire à court terme (STM) et une mémoire à long terme (LTM) » (Roy and Pentland, 2002). L'interaction enfant-adulte est placée dans la mémoire à court terme. Chaque entrée contient un enregistrement vocal lié à une image. Si « suffisamment de motifs vocaux sont répétés avec le même contexte visuel » (Roy and Pentland, 2002) ils sont alors placés dans la mémoire à long terme. Si plusieurs paires motif-images relativement semblables sont présentes dans la mémoire à long terme, le système en fait alors un « item lexical ».

Ce système permet donc de découvrir des mots à partir du signal et d'une image. Toutefois, nous pouvons noter que cette étude ne permet pas de pleinement retranscrire le côté multimodal du langage. En effet, les images utilisées sont en nombre limité et leur représentation est idéalisée.

3 Corpus

Dans le cadre du projet BULB (Breaking Unwritten Language Barrier) plusieurs corpus de langues en danger ont été constitués : 50 heures d'enregistrement en Mboshi, 44 heures en Myene[9] et 40 heures en Basaa[10]. Ces corpus sont constitués d'enregistrements audio, et une partie du corpus en Mboshi est passée par la procédure de *respeaking* (Adda et al., 2016). Une partie des enregistrements en Mboshi a également été traduite en français et transcrite (Godard et al., 2016).

[9] Langue en danger parlée au Gabon

[10] Langue en danger parlée au Cameroun

Il existe également deux sites qui mettent à disposition des corpus de langues en danger. Le premier est français et est géré par le laboratoire LACITO. Ce projet se nomme Pangloss et héberge des corpus de 132 langues différentes provenant du monde entier.[11] Les corpus présentés sont assez hétérogènes aussi bien par leur taille que par leur contenu. En effet, certains corpus comptent plusieurs heures d'enregistrements quand d'autres n'en comptent que quelques minutes. Certains de ces enregistrements sont transcrits, traduits et annotés ; toutefois ce n'est pas le cas de tous les enregistrements. Le contenu des corpus est très divers puisqu'ils sont constitués d'enregistrements d'histoires, de listes de mots ou de chants.

Le second site qui héberge des corpus de langues en danger est le site du DOBES (Documentation of endangered languages)[12]. Ce projet est géré par le *Max Planck Institute for Psycholinguistics*. Le site regroupe des corpus de 67 langues différentes. Les corpus sont plus diversifiés que ceux du projet Pangloss puisque les corpus du DOBES sont constitués d'enregistrements audio, de captations vidéo ainsi que de photographies et de dessins. Toutefois, les corpus ne sont pas librement accessibles dans leur entièreté. En effet, certaines parties des corpus sont en accès privé et il est nécessaire de demander un accès spécial afin de pouvoir les consulter.

4 Conclusion

Le corpus en Mboshi, constitué dans le cadre du projet BULB (Adda et al., 2016), a été utilisé pour tester la découverte non supervisée de mot. Le test avait pour objectif de tester les différents algorithmes de segmentation. Plusieurs approches ont été testées : une approche monolingue utilisant des symboles phonétiques, une approche monolingue utilisant des graphèmes et une approche bilingue utilisant des symboles phonétiques (Godard et al., 2016)

Toutefois les approches employées sont toutes des approches monomodales, ne travaillant que sur du signal vocal transcrit. À notre connaissance, aucune recherche n'a été menée sur la découverte de lexique à partir d'un corpus multimodal pour les langues en danger. C'est la raison pour laquelle nous souhaitons utiliser dans le cadre de nos recherches un corpus multimodal afin de pouvoir extraire du lexique. Les deux modalités qui nous intéressent sont la vision et la parole. En effet, la plupart des langues en danger ne disposent d'aucun système d'écriture. En utilisant la parole, nos recherches pourront être appliquées à toutes les langues en danger. Nous souhaitons également utiliser la

[11] http://lacito.vjf.cnrs.fr/pangloss/ [consulté le 24 mai 2017]
[12] http://dobes.mpi.nl/projects/ [consulté le 24 mai 2017]

vision. En effet, les linguistes se servent régulièrement d'images lors de l'élicitation afin de récolter de la parole.

Comme mentionné précédemment, il existe peu de corpus librement accessibles sur les langues en danger. De plus, lorsque ces corpus existent ils ne sont constitués que d'enregistrements audio avec leur traduction et sont donc la plupart du temps monomodaux. Ainsi, il nous faudra constituer nous même un corpus afin de pouvoir tester les performances à grande échelle d'un algorithme de découverte non supervisée de lexique.

Notre première tâche consistera donc à créer un corpus multimodal afin de simuler une langue en danger. Pour ce faire, nous utiliserons des images provenant d'une base de données (MSCOCO) disposant de légendes en anglais. Ces légendes seront ensuite synthétisées en anglais grâce au logiciel *text-to-speech* développé par Voxygen[13] (conformément à l'approche adoptée par (Bérard et al., 2016)). Nous disposerons donc d'un corpus constitué d'une image avec plusieurs signaux vocaux en anglais, langue qui viendra simuler une langue en danger.

La seconde étape de notre travail consistera à utiliser une approche automatique pour extraire automatiquement du lexique à partir des signaux de parole. Pour ce faire, nous utiliserons de l'alignement temporel dynamique segmental (*Segmental dynamic time warping*) et utiliserons la version optimisée de (Jansen and Durme, 2011) de l'algorithme originellement conçu par (Park and Glass, 2005).

La dernière étape de notre travail de recherche consistera à valider notre protocole sur de vraies données, c'est-à-dire des données qui ont été obtenues par un linguiste sur le terrain. Dans notre cas, nous utiliserons des signaux de parole en Mboshi, obtenus par élicitation à partir d'images.

[13] www.voxygen.fr [consulté le 24 mai 2017]

Chapitre 2

–

Méthodologie de constitution d'un corpus multimodal de très grande taille

Comme spécifié dans la première partie de ce mémoire, il n'existe que très peu de corpus librement accessibles sur les langues en danger. De plus, nos recherches supposent que les corpus que nous pourrions trouver soient multimodaux, incluant des images ainsi que des signaux de parole obtenus par élicitation. Ceux-ci sont donc d'autant plus rares, et lorsque ceux-ci existent, ils sont d'une taille relativement restreinte. C'est pourquoi il nous a fallu **construire un corpus permettant de simuler une collecte à grande échelle de parole obtenue par élicitation à partir d'images** — comme cela est fait lors d'études sur le terrain pour documenter une langue en danger, par exemple.

Nous détaillerons dans ce chapitre la méthodologie que nous avons suivie afin de construire un tel corpus.

1 Corpus multimodaux existants pour les langues en danger

Afin que notre corpus simule au mieux celui d'une langue en danger, il nous a d'abord fallu étudier les spécificités que présente ce genre de corpus. Nous disposions pour ce faire des trois corpus suivants, tous d'une taille très restreinte :

- Extrait du corpus en Mboshi récolté en 2016 par Guy-Noël Kouarata au Congo-Brazzaville (Blachon et al., 2016) ;
- Corpus en Tima récolté entre 2007 et 2012 par G. J. Dimmendaal et son équipe au Soudan. Ce corpus est en partie accessible sur le site de DOBES[14] ;
- Corpus en Tabaq également récolté par G. J. Dimmendaal et son équipe au Soudan. Celui-ci est disponible sur le site du SOAS[15].

1.1 Corpus en Mboshi

L'extrait du corpus en Mboshi dont nous disposions était constitué de 32 images et de 31 fichiers audio obtenus par élicitation grâce à l'application LIG-Aikuma. L'ensemble des fichiers audio représente 41 minutes de parole.

À chaque fichier audio est associé un fichier de métadonnées JSON. Celui-ci contient des informations sur l'appareil ayant servi à l'enregistrement (marque, modèle), sur le fichier WAV enregistré (fréquence d'échantillonnage, nom du fichier WAV, nombre de canaux, durée, date

[14] http://dobes.mpi.nl/projects/tima/project/ [consulté le 24 mai 2017]
[15] https://elar.soas.ac.uk/Collection/MPI143018 [consulté le 24 mai 2017]

d'enregistrement) et sur le locuteur enregistré (nom, sexe, langue maternelle, région d'origine)[16]. Lorsque l'enregistrement est obtenu par élicitation — comme c'est le cas ici — un fichier *linker* est créé afin de lier l'enregistrement à l'image ayant été montrée au locuteur.

Les enregistrements comportent plusieurs locuteurs. Sur l'ensemble du corpus, ils sont au nombre de trois : un homme et deux femmes. Tous les locuteurs n'interviennent pas forcément ensemble sur un enregistrement, parfois il n'y a qu'un homme et une femme, ou dans d'autre l'ensemble des trois locuteurs.

Les fichiers audio sont relativement inégaux en durée, certains faisant moins d'une minute quand d'autres dépassent 2 minutes. En moyenne, les enregistrements sont d'une durée de 1 minute 30.

On peut noter que les locuteurs ne se coupent que très rarement la parole et ont tendance à parler chacun à son tour. La qualité des enregistrements est inégale. En effet, un des locuteurs étant plus éloigné, on l'entend moins nettement que les autres. De plus, des bruits parasites ont également été enregistrés : pleurs d'enfants, chants d'oiseaux, sirènes. En plus de ces bruits parasites, les enregistrements ne sont pas uniquement constitués de parole. En effet, il y a aussi de la non-parole, c'est-à-dire des « stimulations complexes ne comportant pas d'information phonologique » (Signoret, 2010) comme des rires et des bruits de bouches. Les enregistrements comportent également un certain nombre de disfluences marquant des hésitations (« hmm », « euh »), la surprise (« ah », « eh ») ou bien encore l'acquiescement (« Mmhmm »)

Les images ayant servi à l'élicitation se rapportent à l'environnement direct des locuteurs et montrent des plantes, fruits, légumes, animaux et objets. Les locuteurs avaient pour tâche de faire des commentaires « tant sur l'image, le paysage, l'arrière plan que sur l'intention des gens sur la photo ou encore l'impression de chacun au vu de ces images. » (Guy-Noël Kouarata)

Les images portent le nom en Mboshi de l'objet principal qu'elles montrent. De ce fait nous avons pu compter, pour chaque enregistrement, le nombre de fois où le nom de l'objet apparaissait. La fréquence d'apparition du nom de l'objet saillant de l'image varie énormément d'une image à l'autre, allant de 0 à de 21 apparitions. En moyenne, le nom de l'objet figuré dans l'image apparaît 6,5 fois. La liste de l'ensemble des images et du nombre d'occurrences est disponible en annexe à la page 113.

[16] Cependant, lorsque l'enregistrement comporte plusieurs locuteurs, le fichier JSON ne contient que les données relatives à un seul locuteur et non à l'ensemble des locuteurs. De ce fait, il nous manquait des informations sur certains locuteurs de ce corpus.

1.2 Corpus en Tima

Le corpus mis en ligne sur la plateforme DOBES indique que celui-ci est constitué de 162 images couplées à leur fichier de parole. Toutefois, l'ensemble n'est pas entièrement accessible, certains liens vers les fichiers étant brisés. Sur l'ensemble du corpus, nous avons pu récupérer 151 fichiers audio et 135 images. Les enregistrements ont été faits par deux locuteurs pour lesquels nous disposons de métadonnées (nom, âge, sexe, niveau d'éducation et métier).

Contrairement au corpus en Mboshi, les fichiers audio ne sont pas liés de manière indépendante à un fichier de métadonnées indiquant qui a été enregistré. Seul un fichier PDF permet de faire le lien entre les images et les fichiers audio, mais celui-ci ne mentionne toutefois pas qui est le locuteur. Il contient cependant la transcription de ce qui a été prononcé ainsi qu'une traduction en anglais.

Comme les images du corpus Mboshi, les images se réfèrent à l'environnement immédiat des locuteurs. Celles-ci sont plus diversifiées que celles du corpus en Mboshi, on peut notamment voir : des groupes de personnes, des gros plans sur une partie du corps, des gros plans sur une partie d'un objet, des images d'objets ou de bâtiments dans leur contexte, des personnes réalisant des actions, de la nourriture, des portraits, et des animaux.

Même si ce corpus est constitué de plus de fichiers audio que le corpus Mboshi, ceux-ci ne représentent qu'environ 10 minutes de parole. Les locuteurs avaient pour tâche de « faire une courte description de ce qu'ils voyaient sur l'image »[17]. Cette tâche produisait des enregistrements très brefs : les fichiers audio ne font en moyenne que 3.8 secondes. De ce fait, les enregistrements ne présentent aucune disfluence.

Contrairement au corpus Mboshi, où les mots clefs sont répétés de nombreuses fois dans un seul fichier audio, les mots clefs de ce corpus sont répartis sur l'ensemble des fichiers audio. Cela est dû au fait que les images figurent souvent des objets similaires (hommes, femmes ou enfants).

1.3 Corpus en Tabaq

Le dernier corpus est restreint puisqu'il n'est constitué que de 12 images et d'un seul fichier audio de 6 minutes. De la même manière que pour le Tima, il avait été demandé au locuteur de décrire ce qu'il voyait sur les images. Les enregistrements n'ont été faits que par un seul locuteur.

[17] https://hdl.handle.net/1839/00-0000-0000-000E-F767-7@view [consulté le 24 mai 2017]

Un fichier texte permet de faire le lien entre les images et ce qui a été dit par le locuteur. Ce fichier contient également des métadonnées indiquant quel chercheur a supervisé l'enregistrement et quel locuteur a parlé.

Le fichier texte contient les étiquettes temporelles (timecodes) du fichier audio afin de savoir à quel moment une image a été décrite. Les timecodes sont fins, puisque la description d'une même image peut être segmentée en plusieurs parties, correspondant aux pauses faites par le locuteur. Les fichiers audio ont également été transcrits et traduits. Les images, quoique peu nombreuses, présentent également des situations variées : groupe de personnes, gros plan sur une partie du corps, animaux et actions.

Bien que de taille très réduite, ce corpus présente également quelques disfluences, marquant la réflexion (« mh ») et la surprise (« ah »).

2 Constats

Bien que les corpus précédemment présentés soient de natures différentes, aussi bien par leur taille que par les méthodes de récolte, ils présentent quelques points communs.

2.1 Images

Tout d'abord, les images sont de natures variées, certaines montrant des objets dans leur contexte quand d'autres montrent seulement une partie d'un objet en gros plan. Certaines montrent également des personnes effectuant des actions quotidiennes.

2.2 Parole

Les enregistrements varient fortement en longueur, certains étant très courts (Tima) quand d'autres sont très longs (Mboshi). Deux raisons sont envisageables pour expliquer cela : la première est que pour le corpus en Tima, on a demandé aux locuteurs de décrire l'image, sans forcément rentrer dans les détails. La seconde est que pour ce même corpus, les locuteurs enregistrés n'interagissent pas entre eux, alors que pour le corpus en Mboshi il y a un véritable dialogue entre les locuteurs. Nous avons donc des corpus de types différents : un corpus où l'interaction entre locuteurs est prégnante et un corpus où les locuteurs parlent de manière isolée.

Les corpus en Tima et Mboshi ont tous deux été enregistrés par plusieurs locuteurs. En Mboshi les locuteurs sont de sexes différents, quand en Tima les deux locuteurs sont des hommes. On peut

également noter que les locuteurs du corpus en Tima sont d'âges différents (une génération d'écart). Le corpus en Tabaq ne comporte quant à lui qu'un seul locuteur. On peut remarquer que les locuteurs du corpus Mboshi et Tabaq ont tous produit des disfluences, que ce soit pour marquer un doute, la surprise ou bien encore l'acquiescement.

Finalement, on peut remarquer que le débit des locuteurs varie d'une image à l'autre ou d'une prise de parole à l'autre. Dans le corpus en Mboshi par exemple, les locuteurs font parfois de longues pauses et adoptent un débit plus lent alors qu'à d'autres moments ils parlent avec un débit rapide. En Tima et en Tabaq les locuteurs ont tendance à parler lentement, en articulant et en faisant des pauses marquées après les groupes de souffle. En Tabaq, le fichier de métadonnées indique que le locuteur répétait les phrases prononcées afin de « s'assurer qu'au moins un des enregistrements était correct, puis ensuite il parlait de manière plus naturelle »[18]. Cela tend à montrer que la tâche de description d'image peut mener à des enregistrements assez artificiels et très contrôlés.

3 Enrichir un corpus multimodal de grande taille : MSCOCO

3.1 Pourquoi créer un corpus de très grande taille

Les corpus de langues en danger étant rares et de taille restreinte il nous est apparu nécessaire de constituer un corpus de grande taille simulant une langue en danger. En effet, si l'on souhaite tester la performance des algorithmes de découverte non supervisée de lexique, il est souhaitable d'avoir un corpus suffisamment grand permettant de faire différents types d'expériences. Il serait notamment envisageable de regarder si l'algorithme est plus performant lorsqu'il s'agit de trouver du lexique chez un seul locuteur, ou bien encore de regarder si l'algorithme permet de trouver plus facilement du lexique si le sexe des locuteurs est identique. La création d'un nouveau corpus permettrait de déterminer les paramètres optimums qu'il faudrait utiliser selon le résultat que l'on souhaite obtenir. Les corpus de langues en danger ne sont pour la plupart ni annotés ni traduits, et il est par conséquent difficile d'évaluer la performance des algorithmes, et de voir par exemple si ceux-ci hypo- ou hyper-segmentent lorsqu'ils identifient du lexique.

De plus, un corpus liant image et parole pourrait servir à d'autres chercheurs travaillant notamment sur la vision par ordinateur, sur la description automatique d'images en langue naturelle

[18] https://elar.soas.ac.uk/resources/0200-A Documention of Tabaq a Hill Nubian language of the Sudan in its sociolinguistic context/0200-20141220/tabaq/annotations/toolbox/Annotations/d12nhkpictures.txt [consulté le 24 mai 2017]

(*captionning*), sur l'apprentissage de représentations dans des espaces multimodaux (notamment la projection d'images et de parole dans des espaces de représentation communs), etc. Un tel corpus pourrait également servir aux chercheurs travaillant sur l'acquisition du langage en liant énoncés de parole et le contexte visuel est fortement lié à l'audition.

3.2 Point de départ : le corpus MSCOCO

MSCOCO est un corpus d'images utilisé pour l'entraînement de système de vision par ordinateur. MSCOCO est l'abréviation de *Microsoft Common Objects in Context*. C'est là toute l'originalité de ce corpus qui s'attache à ce que les objets soient présentés dans leur contexte. En effet, la plupart des corpus d'images présentent une vision stéréotypique des objets où ils sont « présentés de profil, de manière dégagée et situés au centre d'une image soigneusement composée » (Lin et al., 2014) Ce corpus cherche à prendre le contrepied en présentant des images « reflétant la composition des scènes de la vie quotidienne » (Lin et al., 2014).

Les images de MSCOCO se subdivisent en trois catégories (Lin et al., 2014) :

1. Les images représentant les objets de façon stéréotypique ;
2. Les images représentant les scènes de façon stéréotypique ;
3. Les images représentant des objets ou des scènes de façon non stéréotypique.

Figure 13 Exemple des différentes catégories d'images (Lin et al., 2014)

Les concepteurs du corpus se sont attachés à faire en sorte que celui-ci contienne une majorité d'images de la troisième catégorie.

La distinction de certaines catégories peut être floue, et cela dépend surtout de l'objet que l'on considère dans l'image. Ainsi, l'image de l'homme qui se brosse les dents (c) est considérée comme étant non stéréotypique. Si l'on considère qu'il s'agit de la photographie d'une salle de bain, celle-ci

est en effet non stéréotypique. Toutefois, si l'on considère l'homme qui se brosse les dents, l'image aurait, à notre avis, plus sa place dans la catégorie (b) : scène stéréotypique.

Comme le spécifie le nom du corpus, il s'agit de récolter des images d'objets de la vie quotidienne (*common objects*). La sélection des catégories a tout d'abord été faite en regardant les « 1200 mots les plus fréquemment utilisés pour parler d'objets visuellement identifiables » (Lin et al., 2014). Cette liste a été enrichie par des catégories données par des enfants de 4 à 8 ans, à qui il avait été demandé de « nommer tous les objets qu'ils voyaient à l'intérieur et à l'extérieur » (Lin et al., 2014). Pour terminer, les auteurs ont abouti à 91 catégories d'objets en sélectionnant les catégories qui apparaissaient le plus fréquemment et qui représentaient le plus d'intérêt pour une tâche de vision par ordinateur. Les auteurs assurent que chacun des 91 types d'objets peut « aisément être reconnu par un enfant de 4 ans » (Lin et al., 2014). Les catégories d'objets sont assez variées : brosse à dents, fourchette, éléphant, voiture, personne, etc.

Ainsi chaque image du corpus MSCOCO contient au moins une instance d'une des catégories sélectionnées. Cependant, il est essentiel de faire remarquer que les images ne contiennent pas que des instances des catégories sélectionnées, mais également de nombreux autres objets. La présence de ces objets n'est pas explicitement mentionnée dans les métadonnées de l'image. Toutefois, il peut arriver que les légendes des images en fassent mention.

À chaque image sont associées cinq légendes en anglais décrivant l'image. Dans de très rares cas, certaines images ont six légendes, d'autres quatre. Toutes les légendes ont été écrites par des humains *via* la plateforme créée par Amazon : *Amazon Mechanical Turk*. Cette plateforme permet de subdiviser une tâche qui pourrait s'avérer pénible à faire par une seule personne en un grand nombre de sous-tâches réalisées par de multiples personnes rémunérées. Les annotateurs avaient les consignes suivantes (Chen et al., 2015) :

- Décrire toutes les parties importantes de l'image ;
- Ne pas commencer leur description par « il y a » ;
- Ne pas décrire les détails peu importants ;
- Ne pas décrire ce qui aurait pu se passer avant ou après que l'image a été prise ;
- Si une personne apparaît, ne pas décrire ce qu'on pense que la personne pourrait dire ;
- Ne pas donner de nom propre ;

- Faire des phrases d'au moins 8 mots.

De telles consignes permettent de faire en sorte que les légendes soient toutes factuelles et décrivent réellement ce qui se passe dans les images. Les légendes sont toutes d'une qualité élevée, puisqu'écrites en langage naturel par des humains.

3.3 Pertinence des légendes

Afin de s'assurer de la pertinence des légendes fournies par les annotateurs, les auteurs ont vérifié l'accord inter-annotateurs. Celui-ci a été évalué de manière automatique grâce aux scores suivants :

- **BLEU$_{1,2,3,4}$** : score (de précision) permettant de mesurer « la cooccurrence de n-grammes entre un candidat et une référence » (Chen et al., 2015) ;
- **METEOR** : score d'alignement entre un candidat et une référence prenant en compte l'utilisation de synonymes et l'utilisation de *stem* (radical) à la place des mots entiers ;
- **ROUGE$_L$** : score (de rappel) permettant de mesurer la ressemblance entre un candidat et une référence selon la plus longue séquence commune ;
- **CIDEr-D** : score permettant de mesurer « le consensus dans la description d'une image grâce à la mesure TF-IDF » (Chen et al., 2015).

Metric Name	MS COCO c5	MS COCO c40
BLEU 1	0.663	0.880
BLEU 2	0.469	0.744
BLEU 3	0.321	0.603
BLEU 4	0.217	0.471
METEOR	0.252	0.335
ROUGE$_L$	0.484	0.626
CIDEr-D	0.854	0.910

Tableau 1 Evaluation du consensus inter-annotateur (issu de Chen et al., 2015 TABLE 1 p.4)

« MS COCO c5 » désigne les images pour lesquelles 5 légendes ont été produites et « MS COCO c40 » désigne 5.000 images pour lesquelles 40 légendes ont été produites.

Les scores BLEU sont faibles, et cela est confirmé par les scores METEOR et ROUGE$_L$. Des scores aussi faibles peuvent s'expliquer par la longueur des phrases comparées qui sont toutes très courtes et ne permettent pas d'avoir beaucoup de mots consécutifs identiques. Ces scores tendent à montrer que les phrases produites sont relativement diversifiées d'un point de vue syntaxique.

Toutefois, le score CIDEr-D révèle un bon accord inter-annotateurs concernant la pertinence des descriptions puisqu'il utilise la mesure TF-IDF :

> *« La mesure TF [term-frequency] permet de donner un poids plus élevé aux n-grammes qui apparaissent fréquemment dans les phrases de référence décrivant une image, alors que la mesure IDF [inverse document frequency] permet de réduire le poids des n-grammes qui apparaissent de manière récurrente dans l'ensemble des descriptions »* (Chen et al., 2015)

Cette mesure a été utilisée pour mesurer la pertinence de chacune des légendes d'une image par rapport à l'ensemble des légendes de toutes les images. On constate que le score est assez élevé, ce qui signifie que les scripteurs de légendes ont utilisé les mêmes mots pour désigner les objets présents dans l'image.

3.4 Disponibilité de MSCOCO

MSCOCO a été séparé en trois parties : ½ entraînement (413.915 légendes pour 82.783 images), ¼ validation (202.520 légendes pour 40.504 images) et ¼ test (379.249 légendes pour 40.775 images) (Lin et al., 2014). Nous n'avons accès qu'aux légendes d'entraînement et de validation. De ce fait, nous ne pourrons utiliser que 616.435 légendes provenant de 123.287 images.

Le corpus est librement téléchargeable sur le site de MSCOCO[19]. Les légendes sont disponibles sous forme d'un fichier JSON. Afin de pouvoir aisément manipuler le corpus, nous avons utilisé une API en Python mise à disposition sur GitHub[20].

3.5 Comparaison entre MSCOCO et les corpus de langue en danger

3.5.1 Images

Les images de MSCOCO se rapprochent des images des différents corpus précédemment présentés. En effet, les images de ces derniers montrent des objets de l'environnement immédiat des locuteurs, relatifs à leur vie quotidienne, en contexte. Cela est également le cas pour les images issues de MSCOCO. La différence majeure entre les images de MSCOCO et celles des autres corpus tient aux objets photographiés. En effet, l'environnement d'un locuteur d'un pays occidental diffère largement de l'environnement des locuteurs soudanais ou congolais. Ainsi, il est normal de constater que

[19] http://mscoco.org/home/ et http://mscoco.org/dataset/#download [consultés le 24 mai 2017]
[20] https://github.com/pdollar/coco [consulté le 24 mai 2017]

certaines catégories de MSCOCO (tel qu'*air dryer* ou *baseball bat*) n'apparaîtront pas dans de véritables corpus de langues en danger.

On peut également constater que les images des corpus des langues en danger correspondent aux trois types d'images présentes dans le corpus MSCOCO. Une comparaison des images des différents corpus peut être consultée en annexe à la page 120.

L'ensemble des images de MSCOCO provient de Flickr[21], qui est un site de partage de photos. Ainsi, une majorité de photos de ce corpus a été faite par des amateurs, tout comme les images de nos corpus de références, prises par des linguistes de terrain et non par des professionnels de la photographie.

3.5.2 Légendes

Les légendes des images de MSCOCO se rapprochent des descriptions faites en Tima, les légendes de ce corpus étant très courtes. Elles sont relativement semblables aussi bien sur le fond que sur la forme aux légendes présentes dans MSCOCO.

Les descriptions d'images en Tabaq sont intermédiaires. En effet, certaines sont courtes – se rapprochant ainsi de ce qui a été fait en Tima – quand d'autres sont plus longues et plus proches de ce qui a été produit en Mboshi (description des paysages, descriptions des intentions).

Légendes de MSCOCO[22]	**Légendes Tima[23] (traduites en anglais)**	**Légendes Tabaq (traduites en anglais)**
a plate that has fruit on top of it.	the girl is roasting coffee	this picture shows surely a Tima mountain
a plate of fruit sits on a table near a bookshelf.	the baobab fruit is is hanging high in the tree *(sic.)*	a small boy is riding on a bicycle, and this is surely a cart
a close up of a plate of fruit with oranges	the groundnuts are on top of the granary	and in this picture, two small girls have a jerrycan, it is just water that they are going to bring

Tableau 2 Comparaison des légendes des corpus MSCOCO, Tima et Tabaq

[21] https://www.flickr.com/ [consulté le 24 mai 2017]

[22] Issues de l'image disponible à http://mscoco.org/explore/?id=415378 [consulté le 24 mai 2017]

[23] https://hdl.handle.net/1839/00-0000-0000-000E-F766-9@view [consulté le 24 mai 2017]

4 Oralisation des légendes de MSCOCO

Les légendes de MSCOCO sont toutes écrites et aucune n'a été enregistrée par quelque locuteur que ce soit. Afin d'avoir un corpus multimodal liant des images à la parole il est nécessaire que les légendes des images soient prononcées.

4.1 Précédentes tentatives pour oraliser des légendes

Pour oraliser les légendes de MSCOCO nous faisons face à deux possibilités : employer de véritables humains pour qu'ils enregistrent les légendes une à une, où employer un système de synthèse de la parole.

La première approche a été choisie à deux reprises par (Harwath and Glass, 2015) et (Harwath et al., 2016). (Harwath and Glass, 2015) ont choisi de faire enregistrer les légendes du corpus Flickr8k par de vrais locuteurs. Ils ont eu recours pour ce faire à la plateforme *Amazon Mechanical Turk*. Ce corpus est composé de 8000 images qui ont chacune 5 légendes – soit 40.000 légendes en tout. (Harwath et al., 2016) se sont orientés vers un corpus différent, le corpus Places205. Contrairement à Flickr8k, les images de ce corpus n'ont pas de légende. Ainsi, les locuteurs (*Turker*[24]) avaient pour tâche « d'enregistrer librement une légende pour chacune des images, en décrivant les objets saillants dans la scène » (Harwath et al., 2016). 120 000 images ont été décrites en suivant cette méthodologie. (Harwath et al., 2016) annoncent qu'ils « prévoient de rendre leur corpus publiquement accessible dans un futur proche ». Cependant, six mois après la rédaction de leur article, le corpus n'est toujours pas disponible au public. Nous n'avons pas non plus trouvé trace de la version audio de Flikr8k.

(Chrupała et al., 2017) ont quant à eux décidé d'utiliser un système de synthèse vocale afin de créer leur corpus. Ils ont synthétisé 8 000 images de MSCOCO grâce au système de synthèse vocale de Google (*Google text-to-speech*). Selon les auteurs, l'utilisation d'un tel système permet de générer de la parole « réaliste et de haute qualité » (Chrupała et al., 2017). Toutefois, l'utilisation d'une telle méthode présente quelques faiblesses, puisque ce système « n'utilise qu'une seule voix, et ne permet pas la modification de la vitesse ou l'ajout de bruit ambiant » (Chrupała et al., 2017).

Dans notre cas, faire enregistrer plus de 600 000 légendes par de vrais locuteurs serait une véritable gageure, quand bien même les légendes sont relativement courtes. Cette option n'était donc clairement pas envisageable tant du point de vue du coup humain que financier. Nous avons donc

[24] Personne travaillant par le biais de la plateforme *Amazon Mechanical Turk*

choisi d'utiliser la synthèse vocale pour créer notre corpus. Afin de pallier les problèmes justement pointés par (Chrupała et al., 2017) nous utiliserons le système de synthèse vocale de la société *Voxygen* qui dispose de bien plus d'atouts que Google *text-to-speech*, comme nous allons le voir dans la section suivante.

4.2 Système de Voxygen

Le système de Voxygen permet de faire de la synthèse vocale par concaténation, contrairement au système de Google qui est un système paramétrique. La synthèse paramétrique « consiste à déterminer un modèle numérique de l'ensemble des productions vocales d'un locuteur. […] Ces modèles sont alors utilisés pour générer le signal de parole »[25]. La synthèse par concaténation procède quant à elle « par juxtaposition de segments de parole préalablement enregistrés »[25]. Une telle approche permet de retranscrire « les richesses et les nuances des sons originaux » (Schwarz, 2007) et rend le texte synthétisé très naturel.

4.3 Solutions aux problèmes pointés par Chrupała et al.

Les légendes étant en anglais, nous utiliserons les voix américaines et britanniques dont dispose Voxygen. Les locuteurs britanniques sont au nombre de quatre : un homme (Paul) et trois femmes (Elizabeth, Judith et Bronwen). Les locuteurs américains sont au nombre de quatre également : deux hommes (Phil et Bruce) et deux femmes (Amanda et Jenny). L'utilisation de huit voix différentes permet de pallier le problème d'une voix unique pointé par (Chrupała et al., 2017).

Afin d'apporter une solution au second problème, à savoir la modification de la vitesse, nous utiliserons le logiciel SOX. Ce logiciel permet de facilement travailler sur des fichiers audio et notamment de modifier leur vitesse grâce à la commande *tempo*. Cette commande permet de modifier la vitesse d'un fichier audio sans en modifier la hauteur (*pitch)*. Afin que la modification de la vitesse n'affecte pas le réalisme des légendes générées, nous avons limité la modification de la vitesse à 10 % de la vitesse originale. Si elle est modifiée, la vitesse sera soit 10 % plus lente, soit 10 % plus rapide.

4.4 Serveur de synthèse de Voxygen

Les premiers tests que nous avons menés ont été faits en utilisant le service en ligne *Voxygen TTS Studio*. Toutefois, au vu du nombre important de légendes à synthétiser, Voxygen a accepté de nous

[25] https://www.voxygen.fr/content/la-synthese-vocale-ou-text-speech [consulté le 24 mai 2017]

fournir un autre produit : *Voxygen TTS Server*. Ce produit permet d'installer directement l'équivalent du *Voxygen TTS Studio* sur n'importe quelle machine. Ainsi, la synthèse se fait directement sur la machine sans nécessiter de faire appel au serveur à distance.

Le serveur que nous avons installé est composé de deux parties : un *layer* TCP ainsi qu'un *layer HTTP*. Le *layer* TCP est le module principal du système puisque c'est cette partie qui est chargée d'analyser le texte à synthétiser, et de générer le fichier WAV. Le *layer* HTTP est lui chargé de contacter le layer TCP et de récupérer le fichier WAV généré. Il aurait été possible de nous passer du *layer* HTTP, toutefois les scripts que nous avions précédemment développés étaient compatibles avec celui-ci – moyennant une adaptation à Python 3.5.

5 Ajout de disfluences

Les phrases prononcées par les locuteurs de nos corpus de référence contiennent de nombreuses disfluences, certaines exprimant le doute, la surprise ou bien encore l'acquiescement. Bien entendu, de par leur caractère écrit, le légendes n'en contiennent pas. Nous allons donc ici nous interroger sur la pertinence de l'ajout de telles disfluences lors de la synthèse vocale en vue de rendre notre corpus plus réaliste.

5.1 Éléments de définition

(Constant and Dister, 2012) définissent les disfluences ainsi :

> *« Les disfluences, phénomène propre à l'oral, ont la particularité de briser la linéarité syntaxique de l'énoncé dans lequel elles apparaissent. Elles constituent une interruption (souvent momentanée, parfois définitive) dans le déroulement de l'énoncé. »*

(Bortfeld et al., 2001) identifient quatre types de disfluences en anglais :

- Les répétitions (« *repeats* ») : le locuteur répète un mot ou un syntagme ;
- Les reprises (« *restarts* ») : le locuteur s'interrompt au milieu d'un mot pour reprendre avec un mot différent ;
- Les interjections (« *fillers* ») : telles que « uh », « um », « ah », etc. ;
- Les expressions d'édition (« *editing expressions* ») : telles que « I mean », « sorry », « oops ».

(Clark and Fox Tree, 2002) identifient plusieurs fonctions aux interjections, dont celles d'exprimer l'état de ses connaissances (« huh », « oh ») ou encore d'exprimer sa surprise (« ah », « hah »). Nous retrouvons ces mêmes phénomènes dans nos corpus de référence. Nous n'avons pas constaté la présence de reprises, de répétitions ou d'expression d'édition dans nos corpus. Toutefois, étant donné que ceux-ci étaient dans une langue étrangère, constater de tels phénomènes s'avère très difficile, voire impossible si l'on n'en est pas locuteur natif.

(Bortfeld et al., 2001) signalent que les interjections peuvent se situer à différents endroits :

> *« Les locuteurs utilisent les interjections pour commencer leur tour de parole, pour le terminer, de manière isolée (l'interjection formant l'ensemble du tour de parole), à l'intérieur des phrases, interrompant ainsi des phrases qui auraient sinon été fluides. »*

Plusieurs raisons ont été avancées par (Bortfeld et al., 2001) pour expliquer la présence des disfluences tout en précisant qu'un même phénomène peut avoir plusieurs causes et fonctions différentes. L'auteur indique notamment que la présence de disfluences peut être liée à une « tâche de planification plus importante » (Bortfeld et al., 2001). Ainsi, une interjection telle que « uh » ou « um » peut signaler que locuteur est peu certain de la réponse qu'il donne ou bien encore qu'il « éprouve des problèmes et demande de l'aide » (Bortfeld et al., 2001). (Clark and Fox Tree, 2002) identifient une autre fonction aux interjections telles que « uh » en précisant que celles-ci ont tendance « à augmenter l'attention [de l'interlocuteur] vis-à-vis du discours qui suit » (Clark and Fox Tree, 2002). (Bortfeld et al., 2001) précisent également que les disfluences peuvent servir à gérer les tours de parole. Cela peut expliquer le fait que les dialogues Mboshi contiennent plus de disfluences que les monologues en Tima et Tabaq.

5.2 Pertinence de l'ajout de disfluences

Ainsi, ajouter des disfluences aux légendes de MSCOCO serait justifié. En effet, la description d'image est une tâche qui demande des efforts de planification et où les locuteurs sont souvent amenés à chercher leurs mots. En situation de dialogue, en plus de signaler l'incertitude du locuteur, elles permettraient d'organiser les tours de parole.

Nous avons donc décidé d'ajouter des disfluences aux légendes de MSCOCO. Toutefois, nous n'ajouterons que des interjections. (Bortfeld et al., 2001) signalent que dans leur corpus les expressions d'édition étaient en petit nombre. Les sujets de leur expérience avaient pour tâche de décrire et de classer des images (images d'enfants et de tangram) selon un certain ordre. Sans être

exactement identique, cela s'apparente à ce qui est demandé lorsqu'un locuteur est sollicité pour une tâche d'élicitation par image, où le locuteur décrira l'image.

Nous avons également écarté les reprises et les répétitions. Tout d'abord elles sont bien moins fréquentes que les interjections : 1.94 et 1.47 respectivement tous les 100 mots, contre 2.56 pour les interjections (Bortfeld et al., 2001). Les reprises seraient compliquées à gérer automatiquement. En effet, comment choisir quel mot couper, et par lequel le remplacer ? Cette tâche serait trop ardue et nuirait à la qualité des légendes, qui rappelons-le, ont été écrites par des humains. Les répétitions seraient également compliquées à gérer. Quels segments serait-il justifié de répéter ? De la même manière, cela pourrait nuire à la qualité des légendes. En effet, il faudrait s'assurer que la répétition soit faite à un endroit plausible d'un point de vue grammatical. De plus, les répétitions changeraient la fréquence d'apparition des mots dans le corpus. Dans ce cas, comment s'assurer que la fréquence des mots soit réaliste ?

5.3 Fréquence des disfluences

Dans une perspective de réutilisabilité du corpus, il n'est pas souhaitable d'ajouter trop de disfluences. En effet, l'ajout de disfluences est nécessaire dans le cadre de nos recherches afin de rendre le corpus plus réaliste. Toutefois, pour d'autres recherches, l'ajout de trop nombreuses disfluences pourrait être un frein à l'utilisation du corpus que nous avons créé.

(Bortfeld et al., 2001) indiquent qu'il y a en moyenne 2,56 interjections tous les 100 mots dans leur corpus. Les légendes des images comportant en moyenne 10,7 tokens, l'ensemble des légendes pour une même image totalise donc une moyenne 53,5 tokens. Si l'on se réfère aux chiffres de (Bortfeld et al., 2001) il faudrait donc qu'il y ait 1.28 interjection sur l'ensemble des 5 légendes. Ainsi, il y aura en moyenne 1 légende présentant une disfluence sur les 5.

(Clark and Fox Tree, 2002) signalent que les locuteurs de leur corpus produisent entre 1,2 et 88,5 interjections tous les mille mots. Si l'on suivait ces chiffres, cela reviendrait à faire en sorte que l'ensemble des 5 légendes contiennent entre 0,05 et 4,425 interjections, soit entre 0% et 88,5 % de légendes disfluentes.

On voit combien les chiffres diffèrent d'un locuteur à l'autre, et cela explique certainement les différences constatées d'un auteur à l'autre. Nous avons donc choisi d'intégrer dans le corpus 30 % de légendes disfluentes. En intégrer plus rentrerait en effet en contradiction avec notre volonté de pouvoir réutiliser le corpus dans d'autres contextes.

5.4 Localisation des disfluences

(Bortfeld et al., 2001) indique que lorsqu'une interjection est présente à l'intérieur d'une phrase, elle se positionne dans la majorité des cas soit dans un syntagme verbal, soit dans un syntagme nominal. Nous n'avons toutefois trouvé aucune information plus spécifique qui indiquerait où précisément pourrait se situer une interjection. Ainsi, nous avons décidé d'ajouter les interjections soit après un déterminant, soit avant un nom. En effet, (Bortfeld et al., 2001) précisent que les interjections peuvent servir à indiquer « des difficultés à retrouver le mot à venir ». Ainsi, il semble plus probable que les interjections soient positionnées avant un mot qui a une charge sémantique forte (adjectif ou nom), plutôt que devant des mots outils.

6 Pipeline complet pour l'oralisation des légendes de MSCOCO

Nous allons détailler dans cette section l'algorithme que nous avons mis en place pour synthétiser les légendes.

```
foreach image
  foreach légende
     locuteur ← rand_select(locuteur) [P(locuteur)=1/8]
      disfluence ← rand_select(disfluence)[P(¬disfluence)=0.7 | P(disfluence)=0.3]

     if disfluence
       position ← rand_select(position disfluence) [P(début|milieu|fin)=0.333]
       disfluence_selectionnée ←rand_select(disfluence | locuteur)
       légende ← ajout_disfluence(légende, position, disfluence_selectionnée)

     fichier_wav ← synthèse_voxygen(légende, locuteur)
     fichier_wav ← rand_select(vitesse) [P(0.9|1.0|1.1)=0.333]

     écrire_métadonnées(légende, locuteur, vitesse, disfluence [, position])
```

6.1 Sélection du locuteur

Pour chaque image du corpus MSCOCO nous récupérons l'ensemble des légendes lui appartenant. Pour chacune des légendes récupérées, nous sélectionnons aléatoirement un locuteur qui prononcera cette légende. Le choix de chacun des locuteurs est équiprobable. Ainsi, le corpus sera équilibré, et chaque locuteur aura prononcé en moyenne le même nombre de légendes que les autres.

6.2 Ajout d'une disfluence

Nous décidons par la suite si nous ajoutons une disfluence ou non. La probabilité de ne pas ajouter de disfluence est de 0,7 et celle d'en ajouter une est de 0,3. Si nous ajoutons une disfluence nous sélectionnons ensuite sa position : soit au début, soit au milieu, soit à la fin de la phrase. Chacune des positions à une chance équiprobable d'être sélectionnée. Ainsi, les légendes qui contiendront une disfluences seront équitablement réparties entre celles qui en on une au début, celles qui en ont une au milieu et celles qui en ont une à la fin.

Si nous avons décidé d'ajouter une disfluence, nous sélectionnons une disfluence en fonction de la position choisie et du locuteur sélectionné. En effet, tous les locuteurs ne sont pas à même de prononcer l'ensemble des disfluences. La synthèse étant une synthèse par concaténation basée sur un corpus, si la disfluence n'est pas présente dans le corpus initial, elle ne sera pas prononcée correctement. Afin d'éviter l'inclusion de disfluences dont la prononciation ne se révélerait pas suffisamment naturelle, nous avons étudié quelles disfluences pouvaient être prononcées par un locuteur à une position donnée. Le tableau en annexe page 124 montre pour chaque locuteur quelles sont les disfluences envisageables pour chacune des positions.

Nous n'avons pas fait figurer de disfluence marquant l'approbation (« MmHmm »). En effet, tous nos tests ont échoué, et aucune voix de synthèse ne donne de prononciation correcte. Cette disfluence aurait pu se révéler intéressante, particulièrement si les légendes d'une même image sont réunies en un seul fichier. Cela aurait pu permettre de donner plus de naturel au dialogue.

Le critère principal pour décider si une disfluence pouvait être prononcée ou non était la prosodie. Si la prosodie était semblable à ce qu'un vrai locuteur aurait pu prononcer, la disfluence était validée, sinon elle était rejetée. Nous avons exclu la prononciation de disfluences indiquant la surprise (telles que « oh », « ah », « um oh », etc.) en fin de phrase, car celles-ci n'étaient pas naturelles. De la même manière, celles-ci ne peuvent pas apparaître au milieu des phrases.

Nom du fichier	Légende sans disfluence	Légende avec disfluence
Début		
73_748185_Jenny_Beginning_1-0	The back tire of an old style motorcycle is resting in a metal stand.	**Ah.** The back tire of an old style motorcycle is resting in a metal stand.
488_68454_Bronwen_Beginning_1-1	A batter goes to hit the ball just thrown to him from the pitcher.	**Um, oh.** A batter goes to hit the ball just thrown to him from the pitcher.
285_668473_Elizabeth_Beginning_1-1	A close up picture of a brown bear's face.	**Um, ah.** A close up picture of a brown bear's face.
Milieu		

241_394357_Jenny_Middle_1-1	A man standing holding a game controller and two people sitting.	A**, uh,** man standing holding a game controller and two people sitting.
143_496462_Judith_Middle_1-1	Several birds are sitting on small tree branches.	Several**, er,** birds are sitting on small tree branches.
Fin		
502_652841_Jenny_End_0-9	A furry, black bear standing in a rocky, weedy, area in the wild.	A furry, black bear standing in a rocky, weedy, area in the wild**, uh.**
359_199256_Judith_End_1-0	A traffic light over a street surrounded by tall buildings.	A traffic light over a street surrounded by tall buildings**, um.**

Tableau 3 Exemple de légendes contenant des disfluences

6.3 Synthèse et perturbation de la vitesse

Avant de synthétiser les légendes, celles-ci sont d'abord normalisées : passage du texte en minuscule, ajout d'une majuscule au début et d'un point à la fin si aucun autre signe de ponctuation final n'est présent.

La légende est ensuite envoyée au système de Voxygen qui nous donne en sortie un fichier WAV. La fréquence d'échantillonnage a été fixée à 16.000 Hz. Cette fréquence d'échantillonnage est suffisante pour faire des traitements de la parole par la suite et permet d'avoir des fichiers plus légers que si la fréquence d'échantillonnage par défaut (48.000 Hz) avait été conservée. La vitesse du fichier peut ensuite être modifiée. Soit elle est conservée telle quelle, soit elle est accélérée ou ralentie de 10 %.

La dernière étape du traitement est de générer un fichier de métadonnées associé au fichier audio.

7 Fichiers audio et métadonnées

Comme nous avons pu le constater précédemment, l'ensemble des corpus dispose de fichiers de métadonnées liés aux fichiers audio. Il nous semble donc important que chacun des fichiers audio soit également lié à un fichier de métadonnées. Cela permettra également de savoir quel est le texte qui a été synthétisé, par quel locuteur, à quelle vitesse, etc.

Nous avons fait en sorte que le nom du fichier WAV à lui tout seul permette de donner suffisamment d'informations sur la légende qui a été synthétisée, sans pour autant avoir à ouvrir le fichier de métadonnées. La convention de nommage suivante a été adoptée :

IDimage_IDcaption_Locuteur[26]_PositionDisfluence[27]_Vitesse[28]

Chaque fichier audio est lié à un fichier de métadonnées JSON qui porte le même nom que le fichier WAV, extension exceptée. Ce fichier – dont un exemple est visible en annexe à la page 125 – donne les informations suivantes :

- ID de l'image à laquelle appartient la légende (*imgID*) ;
- ID le la légende synthétisée (*captionID*) ;
- Nom du fichier WAV (*wavFilename*) ;
- Le nom du locuteur (*speaker*) ;
- La durée du fichier WAV (*duration*) ;
- La modification de la vitesse (*speed*) ;
- Le texte de la légende (*synthesisedCaption*) ;
- La position de la disfluence et sa valeur (*disfluency*) ;
- Le *timecode* de chacun des mots, syllabes et phonèmes synthétisés (*timecode*).

Les *timecode* sont directement fournis par le système de Voxygen. Nous avons décidé de conserver cette information qui peut s'avérer précieuse pour de nombreuses tâches. Notre recherche portant sur la découverte non supervisée de lexique, nous pourrons facilement vérifier si le « token » découvert correspond à un mot, ou seulement à une portion de mot.

8 Réalisme du corpus

Les légendes du corpus ayant été oralisées par un système de synthèse (t*ext-to-speech*), il est naturel de s'interroger sur le réalisme et la qualité d'un corpus ainsi créé. La première remarque que nous pouvons faire est que notre corpus comporte beaucoup moins de locuteurs que les corpus créés par (Harwath and Glass, 2015) et (Harwath et al., 2016) qui comportent respectivement 183 et 1163 locuteurs. Toutefois, ceux-ci ne sont pas disponibles librement. Le nôtre n'en comporte que huit. Avoir si peu de locuteurs permet-il d'avoir assez de variations ?

[26] Une valeur possible parmi : Paul, Bronwen, Judith, Elizabeth, Bruce, Jenny, Amanda et Phil
[27] Une valeur possible parmi : Beginning, Middle et End
[28] Une valeur possible parmi : 0-9, 1-0 et 1-1

Nous avons donc procédé à des mesures afin de nous assurer de ce fait, et avons calculé la variabilité interlocuteurs et la variabilité interlocuteurs grâce à des mesures de *dynamic time warping* (DTW). Le DTW permet de « mesurer la similarité de deux énoncés directement au niveau acoustique » (Park and Glass, 2005).

Nous avons sélectionné 99 légendes (visibles en annexe à la page 126) appartenant à 20 images que nous avons fait synthétiser par les huit voix de Voxygen et par la voix de Google *text-to-speech*. Afin d'avoir un point de comparaison avec une voix naturelle, nous avons également enregistré chacune des légendes avec notre propre voix. Nous avons ensuite sélectionné pour l'ensemble des 5 légendes d'une image, un mot qui était répété dans des contextes plus ou moins différents. La portion de signal correspondant à ce mot à ensuite été extraite *via* un script Praat.

Les mesures de DTW ont été faites grâce à deux modules Python. Le premier, *Librosa*, a permis d'obtenir les vecteurs de paramètres MFCC. Le second, *DTW*[29], a permis de faire le calcul de DTW en utilisant comme mesure de coût la distance euclidienne minimale entre deux points des vecteurs d'entrées.

8.1 Variabilité intra- et inter-locuteurs

Nous avons extrait les paramètres MFCC de chacun des segments et avons fait un calcul de DTW entre tous les segments d'une même image d'un locuteur et tous les autres segments appartenant à la même image d'un autre locuteur, et ce pour tous les locuteurs. Nous avons ensuite fait la moyenne des valeurs de DTW de toutes les images pour chacune des paires de locuteurs. Nous avons également comparé entre eux tous les segments d'une même image d'un même locuteur.

	Amanda	Bronwen	Bruce	Elizabeth	Jenny	Judith	Paul	Phil	William	gTTS
Amanda	38,28	60,80	66,10	62,00	53,30	61,67	69,43	68,93	70,41	69,67
Bronwen	60,80	37,39	61,75	52,89	52,84	57,34	57,06	62,23	72,19	63,77
Bruce	66,10	61,75	38,35	59,34	54,68	64,65	57,23	54,83	77,02	69,71
Elizabeth	62,00	52,89	59,34	34,04	53,17	56,13	58,87	62,44	71,83	65,27
Jenny	53,30	52,84	54,68	53,17	38,30	56,00	61,02	60,05	69,21	62,28
Judith	61,67	57,34	64,65	56,13	56,00	47,49	64,94	67,59	72,08	64,16
Paul	69,43	57,06	57,23	58,87	61,02	64,94	40,54	60,37	73,73	68,41
Phil	68,93	62,23	54,83	62,44	60,05	67,59	60,37	45,57	79,60	75,38
William	70,41	72,19	77,02	71,83	69,21	72,08	73,73	79,60	46,47	76,74
gTTS	69,67	63,77	69,71	65,27	62,28	64,16	68,41	75,38	76,74	45,77

Tableau 4 Tableau de variabilité inter- et intra-locuteurs

[29] Module créé par Pierre Rouanet et disponible à l'adresse suivante https://github.com/pierre-rouanet/dtw [consulté le 24 mai 2017]

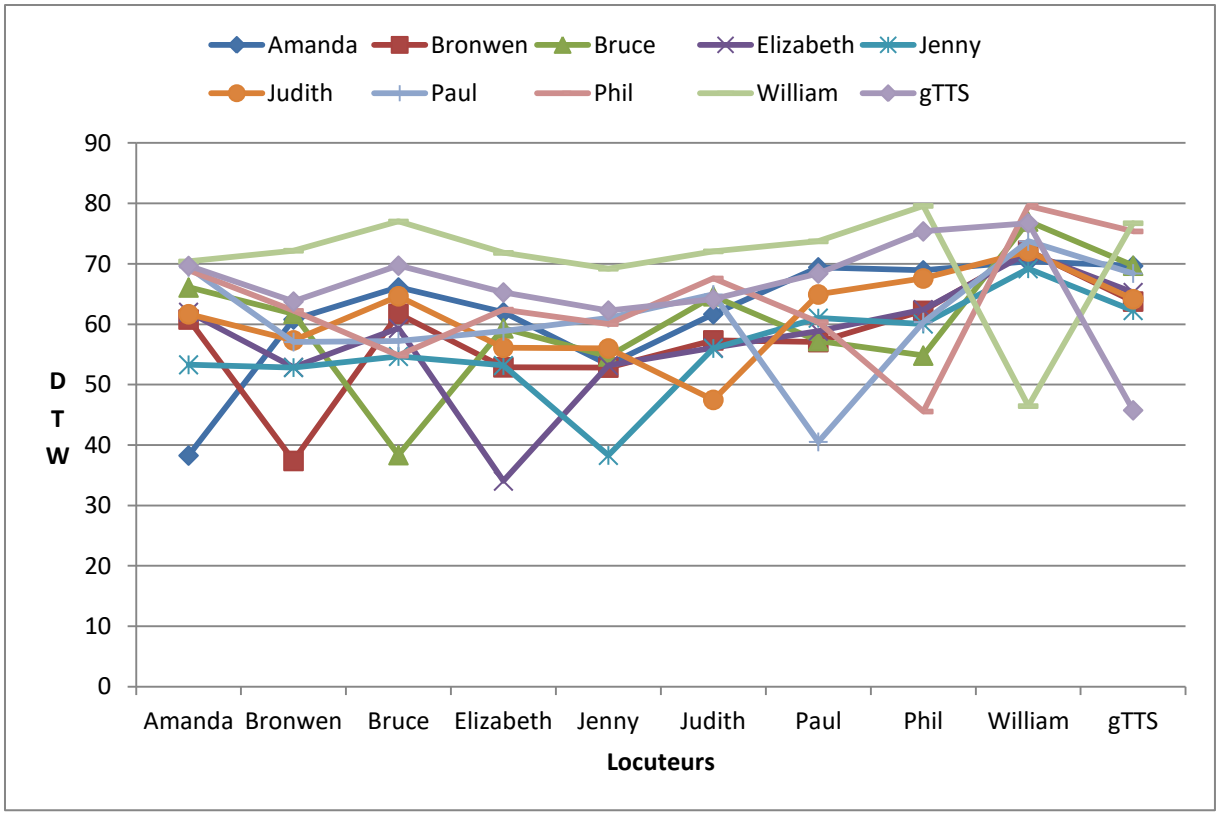


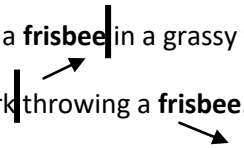
Figure 14 Graphique de la variabilité intra- et inter-locuteurs

La première chose que l'on peut remarquer est que lorsque l'on compare entre eux tous les segments d'une même image d'un même locuteur, les valeurs de DTW sont les plus faibles (diagonale verte du tableau, matérialisée par les minimales de chacun des locuteurs du graphique). Cette remarque est valable pour l'ensemble des voix, que ce soit celles de Voxygen, celle de Google ou notre propre voix. Cela signifie que la variabilité intra-locuteur est donc plus faible que la variabilité inter-locuteurs, ce qui est tout à fait logique.

On pourrait s'interroger sur le fait que la variabilité intra-locuteur ne produise pas des valeurs de DTW nulle. En effet, intuitivement, on pourrait s'attendre à constater ce phénomène puisque l'on compare les mêmes mots d'un même locuteur, générés par un système de synthèse par corpus. Toutefois, ce n'est pas le cas, car la prononciation d'un même mot varie chez un même locuteur, et ce pour deux raisons :

- Le mot n'appartient pas forcément au même bloc prosodique qu'une autre de ses occurrences. Ainsi, si l'on prend ces deux phrases contenant le mot *frisbee*, on constate que le mot n'occupe pas la même position dans la phrase, et appartient par conséquent à deux blocs prosodiques différents :

- A young boy throwing a **frisbee** in a grassy field.
- A young boy in the park throwing a **frisbee**.



Dans le premier cas, le mot *frisbee* est situé à la fin d'un bloc prosodique qui n'est pas situé en fin de phrase, de ce fait, la voix a tendance à monter afin d'annoncer que la phrase n'est pas finie. Dans le second cas, *frisbee*, est situé en fin de bloc prosodique lui-même situé en fin de phrase. De ce fait, la voix à tendance à baisser afin d'annoncer que la phrase est terminée.

- Le mot apparaît avec un contexte gauche et un contexte droit différents.
  - A car driving through a **tunnel** under buildings
    ə kɑr ˈdraɪvɪŋ θru **ə** ˈ**tʌnəl** ˈ**ʌ**ndər ˈbɪldɪŋz[30]
  - Car passing through a very small **tunnel** in a city street.
    kɑr ˈpæsɪŋ θru ə ˈvɛri smɔ**l** ˈ**tʌnəl ɪ**n ə ˈsɪti strit.[30]

  Si l'on prend par exemple des deux occurrences du mot *tunnel* dans les phrases précédentes, on peut constater que les sons à gauche et à droite des deux occurrences du mot sont différents. Ainsi, le phénomène de coarticulation aura tendance à rendre les deux prononciations légèrement différentes, à cause de l'influence des sons adjacents.

A cela, viennent s'ajouter les perturbations de vitesse — ralentissement et accélération — que nous avons implémentées. On peut constater que notre voix est celle qui produit les écarts de DTW les plus forts par comparaison aux autres voix : elle présente le plus de variabilité inter-locuteurs. Toutefois, la variabilité inter-locuteurs des autres voix, même si elle est inférieure à la nôtre, n'est pas négligeable. Notre corpus, même si composé uniquement de voix synthétiques, présentera donc une variabilité relativement semblable à ce que l'on aurait pu obtenir avec des vraies voix convoquées pour une tâche d'oralisation — elles aussi très artificielles — via la plateforme *Amazon Mechanical Turk* par exemple.

De plus amples détails sur la variabilité intra-locuteur mot par mot sont disponibles en annexe à la page 129.

[30] Transcription effectuée automatiquement par http://lingorado.com/ipa/ [consulté le 24 mai 2017]

9 Statistiques sur le corpus créé

La synthèse de l'ensemble du corpus de MSCOCO (train2014 et val2014) a duré six semaines sur un serveur de type PC de bureau[31] : 202 654 légendes pour val2014 et 414.113 légendes pour train2014 ont été synthétisées. L'ensemble des fichiers WAV créés représente un peu plus de 604 heures de parole. L'ensemble des fichiers, audio et métadonnées, occupent un espace de 75 Go environ.

En moyenne, les légendes de l'ensemble du corpus comportent 10,79 tokens et la durée moyenne du fichier WAV généré est de 3,52 secondes.

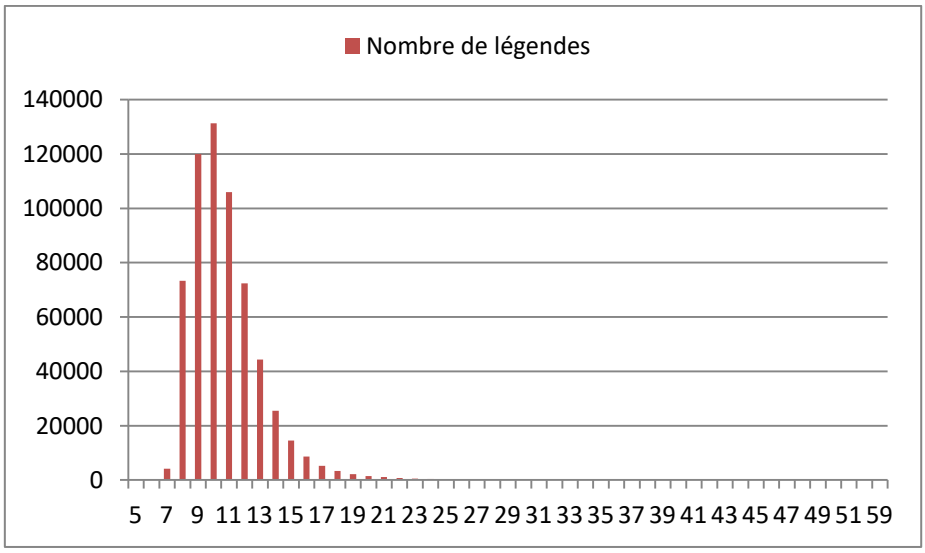


Figure 15 Nombre de tokens par légende

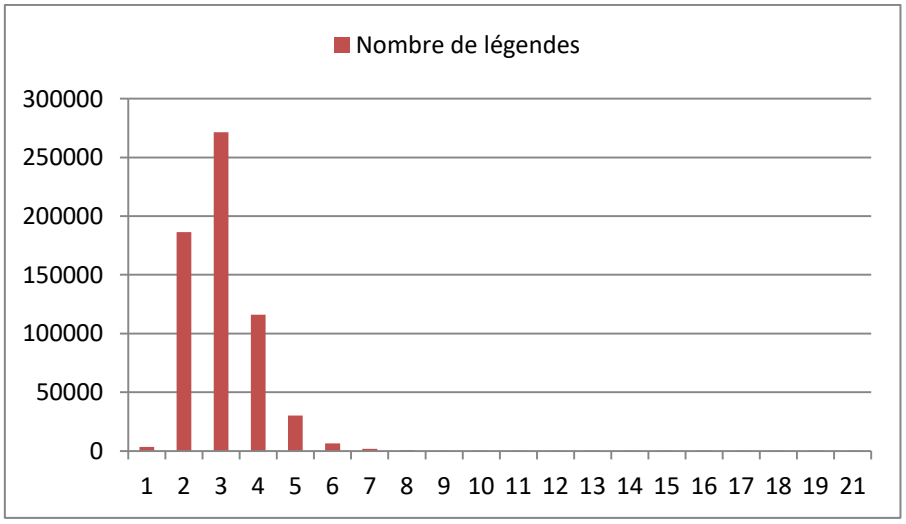


Figure 16 Durée des fichiers WAV

[31] Sur un ordinateur Ubuntu 16.04 64 bits, RAM : 3.8 Gio, Processeur : Intel Core 2 Duo @ 2.93GHz x 2, Carte Graphique Intel G41, Disque Dur 2 To

Comme le montrent les deux graphiques Figure 15 et Figure 16 certaines légendes ont un nombre très élevé de tokens. Toutefois, elles sont en nombre anecdotique : le nombre de légendes ayant un nombre de tokens supérieur ou égal à 20 représente moins de 1% du corpus. De la même manière, les légendes ayant une durée de plus de 6 secondes représentent 1,5% du corpus seulement.

Nous avons également souhaité regarder la diversité lexicale de notre corpus. Pour ce faire nous utilisons la mesure *Type Token Ratio* (*TTR*). Plus cette valeur sera élevée, plus le texte aura une grande variété lexicale. Lorsque cette valeur est faible, cela signifie que le lexique est peu varié et particulièrement prédictible, c'est-à-dire que l'entropie est faible. Afin de pouvoir comparer le TTR de notre corpus, nous l'avons également calculé sur le corpus Europarl ainsi que sur un corpus de textes d'articles de journaux. Le TTR est calculé de la manière suivante :

$$TTR = \frac{nombre\ de\ types\ de\ tokens}{nombre\ de\ tokens} \times 100$$

Le TTR de notre corpus est de 0,40%, celui du corpus Europarl est de 0,18% et celui du corpus d'articles de journaux est de 1,67%. Le TTR de notre corpus est donc plutôt faible, mais cela n'est pas surprenant. En effet, les légendes se rapportent à des images qui contiennent au moins un objet d'une des 91 catégories de MSCOCO. De ce fait, le lexique utilisé par les scripteurs était très orienté, et laissait peu de place à la variété.

10 Points à améliorer

10.1 Synthèse

Nous avons découvert – après le lancement de la synthèse des légendes – que notre synthèse aurait pu être faite de manière plus fine. En effet, le système de Voxygen est compatible avec le langage SSML (*Speech Synthesis Markup Language*).

Lorsque nous utilisions *Voxygen TTS Studio* la seule option dont avions connaissance pour faire synthétiser du texte était d'envoyer un texte brut. De ce fait, lorsque nous avons pu installer un serveur sur notre propre machine, nous avons conservé ce système. Ce langage aurait permis d'avoir un contrôle plus fin sur le texte synthétisé.

Ainsi, les balises `<emphasis>`, `<prosody>` et `<phoneme>` permettent de contrôler avec précision la prononciation. `<emphasis>` permet de mettre de l'emphase sur les mots situés à l'intérieur de celle-ci. `<prosody>` permet de jouer finement sur la prosodie, en permettant

notamment de contrôler le débit et la hauteur de la voix de certaines portions de phrase. `<phoneme>` permet finalement de synthétiser une suite de phonème. Cette balise aurait notamment permis de synthétiser les disfluences que les locuteurs n'arrivent pas à prononcer comme « MmHmm ».

10.2 Disfluences

Il arrive que certaines disfluences au milieu ne soient pas correctement ajoutées comme dans les exemples suivants :

- « Three laptops stacked **on**, **huh**, **top of** each other, from largest to smallest with a cell phone on the top. »[32]
- «A man standing on a **tennis**, **um**, **court** holding a racquet. »[33]

Le problème est lié à TreeTagger qui n'identifie pas correctement les expressions polylexicales. En effet, « on top of » et « tennis court » devraient être considérés comme un tout, ce qui n'est pas le cas ici.

Nous avons également mentionné en annexe à la page 132 les problèmes que nous avons rencontrés lors de la synthèse du corpus.

11 Manipulation du corpus

Afin de pouvoir naviguer dans le corpus, nous avons créé un script Python permettant de facilement le manipuler.[34]

Nous avons fusionné l'ensemble des fichiers de métadonnées JSON dans une base de données au format SQLite. Avoir une base de données SQL est bien plus pratique pour parcourir le corpus que des fichiers JSON indépendants et permet à l'utilisateur de formuler toutes les requêtes qu'il souhaite. Notre script Python fait l'interface entre la base de données et l'utilisateur. L'utilisateur peut ainsi sélectionner les légendes appartenant à un locuteur en particulier, contenant un mot particulier ou bien encore les légendes dont la vitesse a été modifiée. L'ensemble des fonctionnalités de notre script est détaillé en annexe à la page 133.

---

[32] 387_280137_Paul_Middle_0-9.wav
[33] 415_316390_Amanda_Middle_0-9.wav
[34] https://github.com/William-N-Havard/cocoWav/blob/master/cocoWav_API.py [consulté le 24 mai 2017]

12 Conclusion

Nous avons présenté dans cette partie la méthodologie que nous avons suivie afin **d'enrichir un corpus déjà existant en lui ajoutant une nouvelle modalité**. Nous avons montré que celui-ci est **réaliste** même s'il n'était uniquement constitué de voix synthétique. De plus, chacun des fichiers audio de **notre corpus peut être aligné précisément au niveau des mots, syllabes et phonèmes**, ce que ne proposent pas les autres corpus précédemment mentionnés.

Chapitre 3

–

Expérimentations

Nous allons dans cette partie présenter les expériences que nous faites en matière de découverte non supervisée de lexique. Nous commencerons par présenter l'**algorithme qui nous a permis d'extraire le lexique de manière non supervisée**, puis nous présenterons les **résultats** que nous avons obtenus **sur le corpus que nous avons créé** ainsi que **sur le corpus en Mboshi**.

1 Découverte non supervisée de lexique (ZRTools)

Nous avons utilisé l'outil ZRTools créé par (Jansen and Durme, 2011). Cet outil permet de faire de l'alignement dynamique segmental (*Segmental Dynamic Time Warping – S-DTW*) comme présenté dans l'état de l'art au point 2.3.4. L'implémentation du S-DTW ici faite est optimisée pour avoir un temps de traitement plus rapide. Nous détaillerons donc dans les parties qui suivent les optimisations apportées par (Jansen and Durme, 2011).

1.1 Locality Sensitive Hashing (LSH)

L'algorithme prend en entrée des vecteurs de paramètres acoustiques (paramètres PLP à 39 dimensions). Les PLP, tout comme les MFCC sont des vecteurs qui représentent les caractéristiques du signal. Les vecteurs PLP permettent toutefois de « minimiser les différences entre les locuteurs tout en préservant les informations vocales importantes. » (Hermansky, 1990).

Les vecteurs PLP sont des vecteurs multidimensionnels et ils peuvent de ce fait engendrer des temps de traitement relativement importants. La première étape de l'algorithme est donc de réduire la taille des vecteurs PLP. Pour ce faire, les auteurs utilisent le *Locality Sensitive Hashing* « qui est une famille d'algorithmes permettant de hacher des vecteurs de caractéristiques de haute dimension [...] en une signature binaire à faible dimension » (Jansen and Durme, 2011). Les signatures binaires peuvent ensuite être utilisées pour calculer des distances de Hamming entre plusieurs paires de signatures. Cette distance permet « d'approximer une mesure de distance dans l'espace originel » (Jansen and Durme, 2011). L'algorithme LSH a donc pour but de rendre les futures opérations de moins lourd les futures opérations de calcul de similarité cosinus entre deux vecteurs.

Il est nécessaire de noter que les algorithmes du type LSH ont pour particularité d'être des fonctions de hachage où l'apparition de collisions est souhaitée. On peut définir le terme collision de la manière suivante :

> « Soient deux objets *X* et *Y*. La probabilité que leur valeur de hachage *h(X)* et *h(Y)* soit la même dépend de la distance entre X et Y. Plus la distance est faible, plus il est probable

que *X* et *Y* entrent en collision grâce à la fonction de hachage *h*. Le fait d'avoir *X* et *Y* tel que *h(X)=h(Y)* est appelé une collision » [35]

Cela permettra de retrouver facilement des portions des vecteurs PLP qui sont similaires, mais pas tout à fait identiques.

1.2 Point Location in Equal Balls (PLEB)

Cet algorithme permet de construire une matrice de similarité approximative à partir des signatures binaires LSH. L'algorithme crée plusieurs matrices de similarité, soit à partir de plusieurs signatures d'un même fichier audio, soit entre des signatures binaires appartenant à des fichiers audio différents.

L'intérêt de cet algorithme est qu'il ne fait de calcul de similarité cosinus sur toutes les paires possibles de signature binaires, mais seulement celles qui sont suffisamment proches. L'algorithme va donc chercher les plus proches voisins approximatifs (*« approximate nearest neighbor search »*) d'une signature binaire donnée et calculer leur similarité cosinus. La sélection des signatures binaires utilisées pour la comparaison permet « d'éviter de perdre du temps dans des régions de la matrice où les valeurs [de similarité] seraient négligeables » (Jansen and Durme, 2011).

L'algorithme procède en deux étapes :

- Les signatures binaires sont triées dans un ordre lexicographique : « cela revient à trier les chaînes binaires selon leur ordre alphabétique » (Jansen and Durme, 2011)

- On parcourt ensuite la liste triée et pour chacun des éléments on calcule une similarité cosinus avec un nombre d'éléments fixe situés en dessous de lui dans la liste.

Afin d'augmenter la performance de l'algorithme, lorsque deux paires de signatures binaires ont une similarité cosinus suffisamment proches, l'algorithme va regarder si les signatures situées avant et après dans le signal de parole présentent également une similarité suffisante. En effet, ce n'est pas parce que deux signatures binaires sont proches qu'elles seront nécessairement dans une portion de signal similaire. Cette opération permet de soulager les traitements ultérieurs en évitant des comparaisons inutiles.

[35] http://searchivarius.org/blog/does-locality-sensitive-hashing-lsh-analysis-has-fatal-flaw [consulté le 24 mai 2017]

Deux signatures sont considérées comme similaires lorsque leur similarité cosinus est supérieure à un certain seuil. Par défaut, celui-ci est fixé à 0,5.

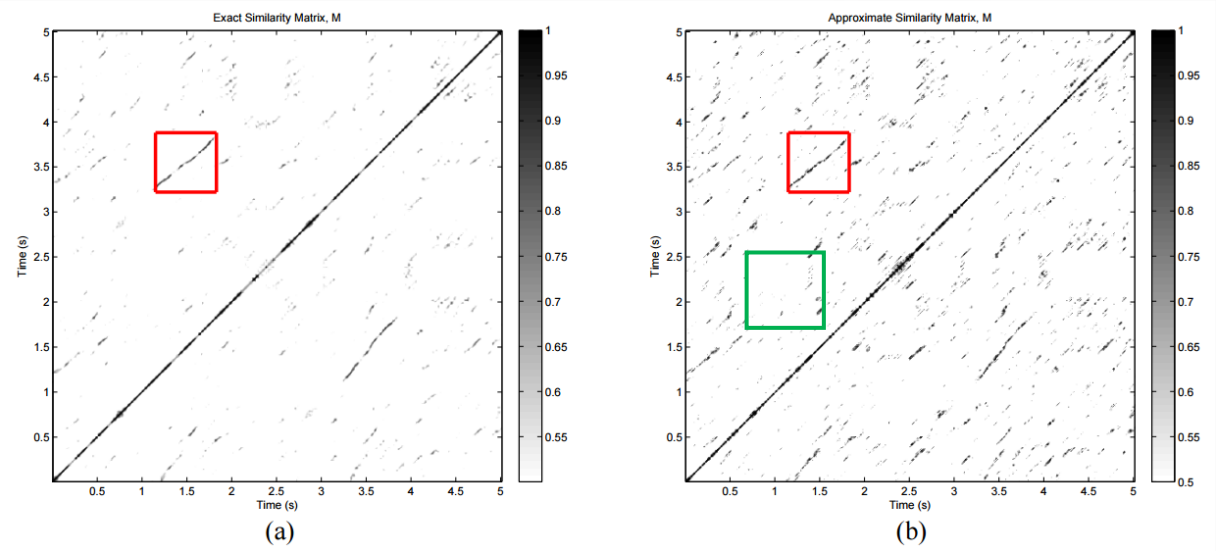


Figure 17 Matrices de similarité exacte (a) et approximative (b) (Jansen and Durme, 2011)

La Figure 17 montre en (a) la matrice de similarité exacte entre deux signaux de parole, et en (b) la matrice approximative construite à partir des algorithmes LSH et PLEB. On constate que la matrice (b) est bien plus bruitée (carré vert), ce qui est normal puisqu'elle résulte d'une suite d'approximations. Toutefois, le bruit n'empêche pas de distinguer des portions de signal qui sont similaires (carrés rouges) qui correspondent aux deux occurrences du mot « recyclable ».

1.3 Recherche de similarité « en deux passes »

Cette étape est la dernière étape de l'algorithme de découverte à proprement parler. Elle consiste à trouver des segments acoustiques similaires en utilisant des mesures d'alignement dynamique (DTW).

1.3.1 Première passe

La première passe va consister à identifier des segments diagonaux dans la matrice (à l'image des carrés rouge de la Figure 17). Cette étape du traitement applique des algorithmes du traitement de l'image, puisqu'il est en effet possible de considérer la matrice comme une image.

Au cours de la La première étape, on retire le bruit en enlevant toutes les points de similarités entre deux signatures qui sont inférieurs à un certain seuil.

La seconde étape consiste à appliquer un filtre gaussien à l'image de la matrice. Ce filtre aura pour effet de flouter l'image en « étalant les segments [diagonaux] restants » (Jansen et al., 2010). Cela

permet de modéliser « une variation dans le débit de parole en autorisant une déviation [des segments] de l'angle hypothétique de 45° [qu'ils devraient avoir] » (Jansen et al., 2010) En effet, plus deux occurrences d'un même message (mot) sont proches et prononcées à la même vitesse, plus elles forment une ligne diagonale inclinée à 45°. Si l'une des deux occurrences est prononcée plus lentement que l'autre, la ligne diagonale qu'ils formeront aura un angle soit plus obtus soit plus aigu. L'application de ce filtre permet donc de trouver des segments similaires qui présentent des variations de vitesse.

La dernière étape consiste à rechercher les lignes diagonales. Cette étape fait appel à une transformée de Hough (Jansen et al., 2010) qui permet de détecter des lignes dans une image.

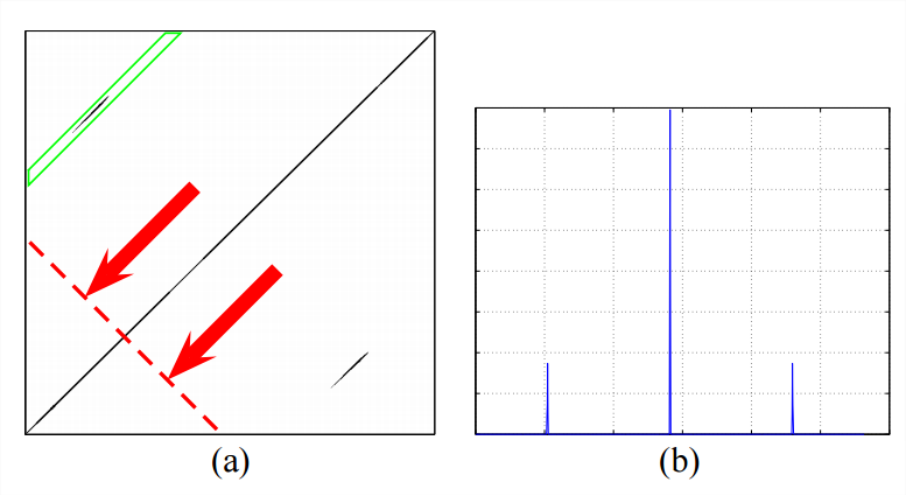


Figure 18 Résultat d'une transformée de Hough (Jansen et al., 2010)

La transformée de Hough fournit en sortie le graphe (b) où chaque pique indique la présence d'une ligne diagonale. Cela permet ensuite de définir des bandes (en vert sur la matrice (a)) où sont présentes des lignes diagonales.

1.3.2 Seconde passe : application de l'algorithme segmental (S-DTW)

La dernière étape du traitement consiste à utiliser une implémentation modifiée de l'algorithme de S-DTW proposé par (Park and Glass, 2005).

(Jansen and Durme, 2011) mentionnent que les variations intra-locuteurs (prosodie, accentuation) et inter-locuteurs peuvent engendrer des déformations des lignes diagonales à 45°. Par conséquent elles peuvent avoir été interprétées comme du bruit et retirées aux étapes précédentes. Leur découverte de segments en deux passes permet toutefois de résoudre ce problème. En effet, on peut « supposer que s'il existe une forte correspondance [entre deux segments et que ceux-ci forment une ligne] déformée non linéairement, elle pourrait encore contenir un segment, non

déformé, de la taille d'une syllabe, que la première passe pourrait découvrir » (Jansen and Durme, 2011).

Lors de l'application de l'algorithme de S-DTW, celui-ci prend pour point de départ le centre d'une ligne diagonale découverte lors de la première passe, et il est contraint de se limiter à un espace de recherche restreint « en limitant l'écart maximal de déviation possible F (en frames) […] de la diagonale ». (Jansen and Durme, 2011). Plus concrètement, cela revient à stopper le processus de S-DTW lorsque le chemin créé dépasse les bandes vertes de la Figure 18 par exemple.

Comme nous le voyons, le S-DTW est appliqué le plus tard possible dans le traitement afin de le rendre moins lourd et plus rapide.

1.4 Clustering

La dernière étape de découverte non supervisée de lexique consiste à regrouper les segments semblables en entre eux selon une valeur de DTW donnée.

Cette étape est réalisée au moyen d'un graphe, où « chaque nœud correspond à un segment acoustique et les arêtes représentent les valeurs de DTW » (Liu et al., 2017). Il suffit ensuite de spécifier une valeur de DTW spécifique pour récupérer seulement les segments dont la valeur de DTW est égale ou supérieure à la valeur spécifiée par l'utilisateur. Ensuite « chaque élément connecté du graphe forme un cluster de segments acoustiques » (Liu et al., 2017).

L'étape de clustering est totalement indépendante des étapes précédentes. Ainsi, il est possible de d'obtenir différents clusters en seuillant avec différentes valeurs de DTW sans avoir à refaire toutes les étapes précédentes.

1.5 Performances

L'alignement dynamique segmental (S-DTW), dans son implémentation de base, est une opération qui est coûteuse puisqu'elle implique de découper l'ensemble des signaux en un certain nombre de xbandes pour ensuite faire des mesures de DTW entre chacune des bandes ainsi créées (voir état de l'art 2.3.4). L'optimisation proposée par (Jansen and Durme, 2011) permet de réduire considérablement le temps de traitement, en effectuant les opérations de S-DTW le plus tard possible sur les segments qui ont le plus de chances d'avoir un score élevé.

(Jansen and Durme, 2011) signalent notamment que leur implémentation parallélisée permet de traiter quatre heures de parole en 2,51 heures-machine. La version de base proposée par (Park and

Glass, 2005) aurait mis près de 528 heures-machine (soit 22 jours) pour traiter quatre heures de parole. Le gain de temps est donc considérable.

2 Expérimentations sur le corpus MSCOCO

Nos premières expérimentations ont été faites sur le corpus MSCOCO dont nous avons détaillé la création au chapitre précédent. Nous commencerons par présenter les métriques d'évaluation que nous avons utilisées puis nous présenterons les résultats que nous avons obtenus sur le corpus MSCOCO.

2.1 Métriques d'évaluation

Les métriques présentées par (Ludusan et al., 2015) se veulent être une « boîte à outils pour l'évaluation des systèmes de découverte de lexique » qui permettent d'évaluer les sorties de ces systèmes à différents niveaux. Nous nous sommes orientés vers cette « boîte à outils » puisqu'elle a été utilisée pour évaluer les systèmes entrant en compétition au *ZeroSpeech2015* et servira pour le *ZeroSpeech2017* [36]. Nous présenterons ici succinctement les différents niveaux d'évaluation.

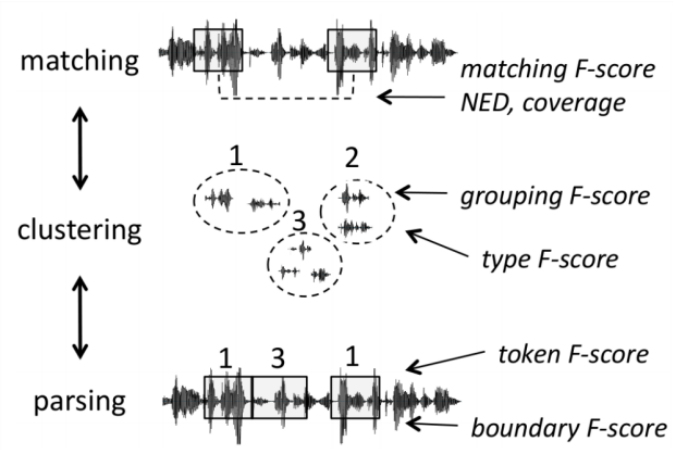


Figure 19 Niveaux d'évaluation proposé par (Ludusan et al., 2015)

2.1.1 Parsing

Le *parsing* désigne le fait d'utiliser les clusters ou les fragments extraits « pour assigner des frontières de segmentation » (Ludusan et al., 2015). ZRTools fait du *parsing* de manière implicite, en effet, les frontières sont implicitement créées à partir du moment où un segment est repéré.

[36] http://sapience.dec.ens.fr/bootphon/ [consulté le 24 mai 2017]

La mesure *token* (précision/rappel/F-mesure) permet d'évaluer si des tokens ont été identifiés parfaitement (sans hypo- ou hypersegmentation) et *boundary* (précision/rappel/F-mesure) permet de voir « combien de vraies frontières de mots on correctement été trouvées» [37].

2.1.2 Clustering

Le *clustering* vise à regrouper les segments semblables entre eux. Deux métriques sont proposées : le *grouping* (précision/rappel/F-mesure) et le *type* (précision/rappel/F-mesure).

Le *grouping* (précision/rappel/F-mesure) évalue « la qualité et l'homogénéité des groupes et des segments découverts [d'un point de vue phonétique] » (Ludusan et al., 2015). *Type* (précision/rapple/F-mesure) permet d'évaluer le *clustering* par rapport aux mots du lexique. En effet, « un système pourrait avoir des clusters très purs, mais avoir systématiquement segmenté de manière incorrecte les mots » [37] *Type* permet donc de comparer les résultats de notre *clustering* à un *clustering* idéal si le corpus avait été segmenté de manière correcte à 100% et tous les mots d'un même type réunis au sein d'un même cluster.

2.1.3 Matching

Le *matching* consiste à créer « des listes de paires de fragments, chacun correspondant à des portions du signal de parole » (Ludusan et al., 2015). Trois outils sont proposés pour évaluer le *matching* : précision/rappel/F-mesure, couverture et la distance normale d'édition (NED).

La distance normale d'édition permet de s'assurer de la « justesse du processus de *matching* et peut être interprétée comme le pourcentage de phonèmes partagés entre deux chaînes » (Ludusan et al., 2015). Elle est calculée en divisant la distance Levenshtein des deux chaînes par la longueur de la chaîne la plus longue. Le site ZeroSpeech2017[37] précise que « lorsque deux chaînes ont la même transcription la distance est de 0, et de 1 lorsque les deux chaînes n'ont aucun phonème en commun ». La couverture permet quant à elle de mesurer « le pourcentage de phonèmes découverts appartenant à des fragments par rapport à l'ensemble des phonèmes présents dans le corpus » (Ludusan et al., 2015). Finalement, le *matching* permet de voir si « l'algorithme est capable de localiser des fragments de paroles identiques dans le corpus » (Ludusan et al., 2015).

[37] http://sapience.dec.ens.fr/bootphon/2017/page_3.html [consulté le 24 mai 2017]

2.2 Résultats

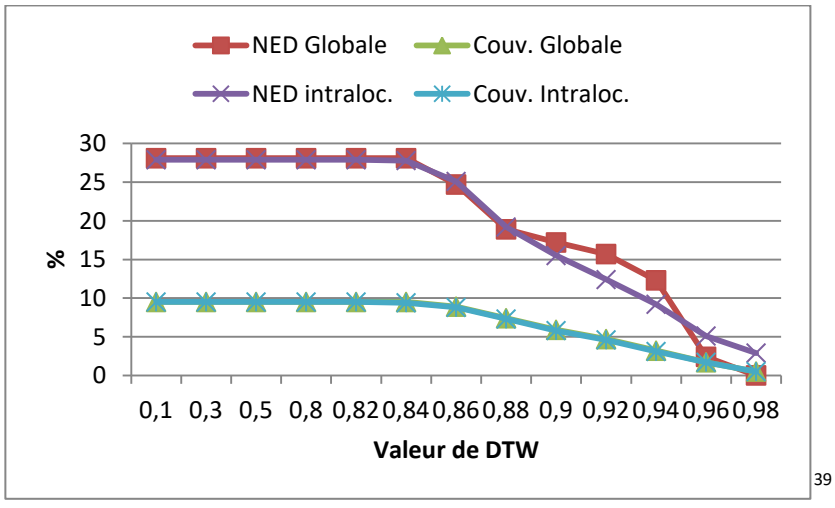
Nous avons utilisé un sous-ensemble de notre corpus MSCOCO : 10 000 légendes appartenant à 1.998 photographies, le tout représentant 9 h 45 de parole et 108 022 mots différents[38]. La découverte non supervisée de lexique a été relativement longue, puisque faite sans parallélisation, et a duré 168 heures, soit une semaine.

Nous avons reporté les résultats que nous avons obtenus à la page suivante. Pour des questions de lisibilité nous avons jugé utile de convertir les résultats en pourcentage.

Le premier élément que nous pouvons remarquer est que les résultats sont strictement identiques pour une valeur de DTW comprise entre 0,1 et 0,82. Cela signifie donc que tous les segments trouvés par ZRTools ont *au moins* une distance de DTW égale à 0,82. Une fois la barre de 0,82 passée, les résultats évoluent très nettement. Cette « non-évolution » des résultats à des valeurs de DTW inférieures à 0,82 s'explique par le fait que ZRTools ne compare des segments de paroles entre eux que s'ils ont une probabilité suffisamment forte d'être proches. A de nombreuses reprises lors du traitement, ZRTools écarte des traitements ultérieurs les segments qui ne respectent pas un certain seuil de similarité.

2.2.1 Matching

Nous remarquons également que plus la valeur de DTW augmente, plus la distance normale d'édition (NED) et la couverture diminuent.



[39]

Figure 20 Evolution de la NED et de la Couverture en fonction de la valeur de DTW

[38] Calcul effectué grâce à la méthode *word_tokenize()* de NLTK en ne conservant que les mots composés de caractères alphanumériques (i.e. ponctuation exclue)

[39] Nous avons exclu du graphique les valeurs obtenues à 0.99 qui présentaient des effets de bord importants

Tableau 5 Résultats de ZRTools obtenus sur notre extension du corpus MSCOCO

Niveau d'évaluation		MATCHING					CLUSTERING						PARSING					
DTW		NED	Couv.	Matching			Grouping			Type			Token			Boundary		
				P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
0,1	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,9	9,5	31,9	0,6	1,2	19,1	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,4	5,9
0,3	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,9	9,5	31,9	0,6	1,2	19,1	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,4	5,9
0,5	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,9	9,5	31,9	0,6	1,2	19,1	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,4	5,9
0,8	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,9	9,5	31,9	0,6	1,2	19,1	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,4	5,9
0,82	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,9	9,5	31,9	0,6	1,2	19,1	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,4	5,9
0,84	global	28,1	9,5	30,2	0,1	0,2	14,8	48,9	22,2	0,5	0,2	0,2	0,3	0	0,1	27,7	3,4	6,1
	intralocuteur	27,8	9,4	31,9	0,6	1,2	19,2	48,8	27,1	0,4	0,1	0,2	0,3	0	0	27,2	3,3	5,9
0,86	global	24,7	8,9	33,2	0,1	0,2	16,5	48,9	24	0,5	0,1	0,2	0,3	0	0	28,3	3,2	5,8
	intralocuteur	25,1	8,8	33,8	0,6	1,2	20,3	49,4	28,4	0,4	0,1	0,1	0,3	0	0	27,9	3,2	5,6
0,88	global	18,9	7,4	39,4	0,1	0,2	18,8	48,3	26,5	0,6	0,1	0,2	0,3	0	0	30,4	2,7	4,9
	intralocuteur	19,2	7,3	40,1	0,6	1,2	24,2	51,9	32,7	0,4	0,1	0,1	0,4	0	0	30,2	2,6	4,8
0,9	global	17,2	5,9	44,9	0,1	0,2	22,8	64,2	32,1	0,6	0,1	0,2	0,4	0	0	32,4	2,1	4
	intralocuteur	15,5	5,8	46,7	0,5	1	28,4	51,7	35,8	0,6	0,1	0,1	0,4	0	0	32	2	3,8
0,92	global	15,7	4,7	48,1	0,1	0,1	24	55,5	29,8	0,3	0	0,1	0,2	0	0	33,4	1,6	3,1
	intralocuteur	12,4	4,6	52,9	0,4	0,8	33,6	51,7	40,3	0,3	0	0,1	0,2	0	0	32,9	1,6	3
0,94	global	12,3	3,2	56,2	0	0,1	35,5	57,7	43,1	0,2	0	0	0,1	0	0	35,7	1,1	2,1
	intralocuteur	9,2	3,1	59	0,2	0,5	41,9	50,7	44,4	0,4	0	0	0,2	0	0	36,2	1,1	2,1
0,96	global	2,4	1,7	90,9	0	0	83,3	100	88,9	0,4	0	0	0,2	0	0	40,3	0,6	1,2
	intralocuteur	5,1	1,7	72,8	0,1	0,2	64,1	73,2	65,8	0,3	0	0	0,3	0	0	40,6	0,6	1,1
0,98	global	0	0,5	100	0	0	100	100	100	0	0	0	0	0	0	42,7	0,2	0,4
	intralocuteur	2,9	0,5	81,4	0	0,1	85,3	78,3	81,2	0	0	0	0	0	0	45,6	0,2	0,3
0,99	global	100	0,1	0	0	0	0	0	0	0	0	0	0	0	0	45,5	0	0,1
	intralocuteur	0	0,1	96,7	0	0	100	50	66,7	0	0	0	0	0	0	54,1	0	0,1

Ces résultats ne sont pas surprenants. En effet, plus la valeur de DTW est élevée, plus la ressemblance entre les segments est grande. Il est donc normal de voir la NED diminuer : cela signifie que les segments trouvés sont très proches phonétiquement. La diminution de la couverture est également attendue : plus l'on est strict sur la ressemblance des segments, moins l'on trouvera de segments proches phonétiquement, et par conséquent, moins l'on aura couvert le corpus. Les valeurs intralocuteurs évoluent de la même manière. On peut toutefois noter que la NED intralocuteur est légèrement meilleure que la NED globale pour une valeur de DTW comprise entre 0,90 et 0,94.

Ces résultats sont confirmés par les valeurs de précisions obtenues au score *matching*.

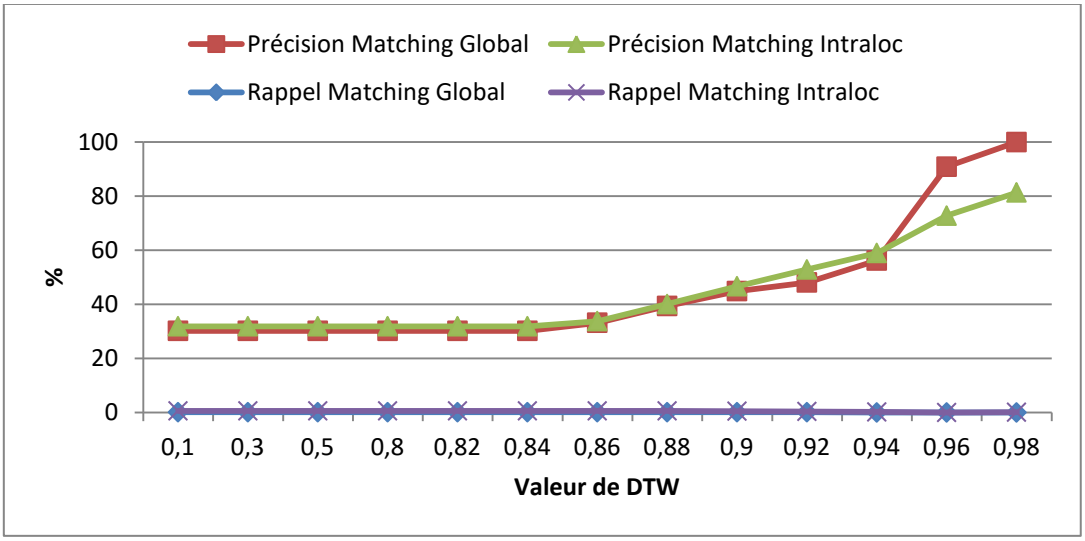


Figure 21 Evolution de la précision et du rappel du *matching* en fonction de la valeur de DTW

Quelle que soit la valeur de DTW choisie, la précision est en permanence supérieure au rappel. Les segments trouvés sont donc relativement proches, mais de nombreux autres segments semblables n'ont pas été trouvés

Etant donné que nous utilisons un corpus synthétique, nous nous serions attendu à trouver beaucoup plus de mots dans un contexte intralocuteurs. Nous pouvons cependant constater que ce n'est pas le cas. Nous pensons que cela est dû à des variations suprasegmentales. En effet, comme nous l'avons mentionné, la variabilité intralocuteur est en grande partie due à des variations de prononciation causées par des variations prosodiques sur les occurrences d'un même mot. Ce phénomène est parfaitement illustré par le Tableau 27 (visible en annexe à la page 130) où les deux occurrences du mot « people » présentent des variabilités intralocuteur différentes. Dans le premier cas (occurrences issues de la légende 257) la variabilité est très forte, ce qui indique que les mots sont prononcés de manière relativement différente. Alors que dans le second cas (légende 524333) la

variabilité intralocuteur est très faible, ce qui montre que la prononciation des différentes occurrences de ce mot est proche. Ainsi, dans le premier cas il sera plus difficile d'identifier les différentes occurrences du mot « people » comme étant similaires.

2.2.2 Clustering

Les mesures de *clustering* et plus particulièrement le *grouping* nous donnent des informations sur la pureté des classes créées.

Plus la valeur de DTW augmente, plus le rappel et la précision augmentent, c'est-à-dire que nos *clusters* sont de plus en plus purs. On constate une nette amélioration du rappel et de la précision lorsque la valeur de DTW est fixée à 0,96. Cette observation est cohérente avec le fait que la NED est également très faible à 0,96. En effet, puisque la distance de Levenshtein entre deux suites phonémiques est faible, les *clusters* ne peuvent en être que plus purs.

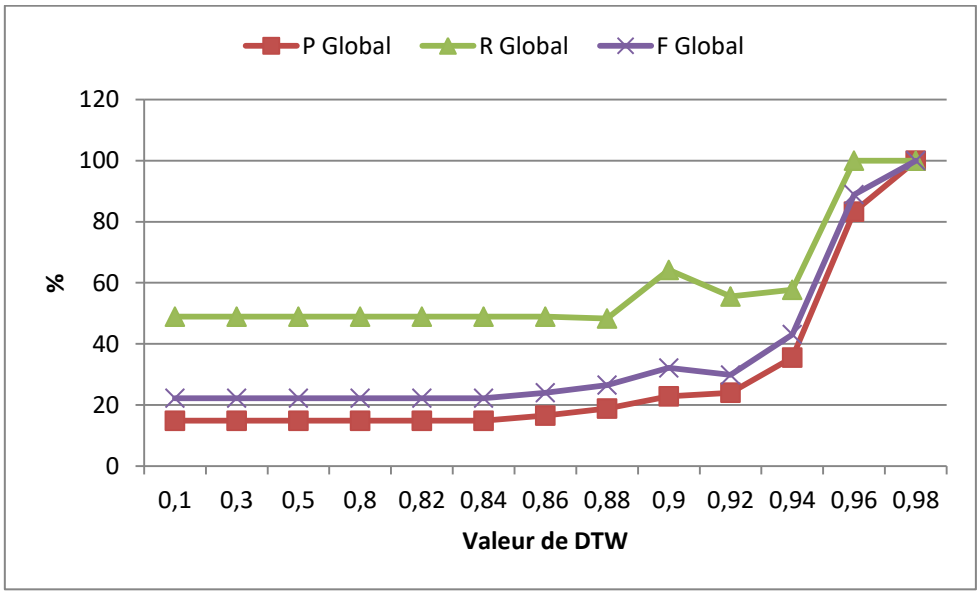


Figure 22 Evolution de la précision et du rappel du *grouping* en fonction de la valeur de DTW

Dans l'ensemble on constate que le rappel est supérieur à la précision. La précision relativement faible tend à montrer que les *clusters* sont bruités et contiennent un ou plusieurs segments relativement différents des autres. *A* contrario, le rappel plus élevé nous indique que le silence est faible, c'est-à-dire que même s'il manque des segments pertinents à l'intérieur des *clusters*, une bonne partie à quand même été correctement attribués à un *cluster*.

Nous pensons que l'augmentation du rappel lorsque la valeur de DTW augmente (particulièrement à des valeurs supérieures à 0,96) s'explique par le fait que beaucoup moins de segments sont trouvés.

Plus la valeur de DTW augmente, plus les segments trouvés entre eux sont proches, ce qui explique le fait qu'il y en ait moins. Ainsi, comme il y a moins de segments trouvés et que ceux-ci sont plus proches sur le plan acoustique, il est beaucoup plus facile de les réunir et par conséquent le silence baisse et le bruit diminue également.

Nous pouvons également noter que la précision obtenue d'un point de vue intralocuteur est meilleure que la précision globale en dessous de 0,94 : cela révèle que le système a plus de facilités à regrouper les segments d'un même locuteur entre eux, plutôt que ceux de plusieurs locuteurs mélangés.

Les valeurs de *type* sont vraiment faibles. Cela montre que la segmentation des mots n'est pas correcte. Soit beaucoup de mots ont été tronqués, soit au contraire, le système a eu tendance à hypo segmenter et considérer un ensemble de mots comme une seule unité.

2.2.3 Parsing

La mesure *token* confirme que peu de tokens de référence ont été correctement retrouvés. Toutefois, le résultat de *boundary* est meilleur. Bien que le système ne soit pas parvenu à identifier beaucoup de tokens, il est quand même parvenu à poser des frontières de manière correcte. La précision n'est néanmoins pas élevée et tend à montrer que certaines des frontières posées sont fausses. Le rappel très faible (inférieur à 3 pour l'ensemble des valeurs de DTW) montre que le système a oublié énormément de frontières.

On ne constate pas de différence notable entre la précision *boundary* globale et intra-locuteurs. Cela est certainement dû au fait que les clusters regroupent principalement des segments appartenant à un seul locuteur. Le nombre de cluster où des segments appartiennent à plusieurs locuteurs sont en nombre plus faible.

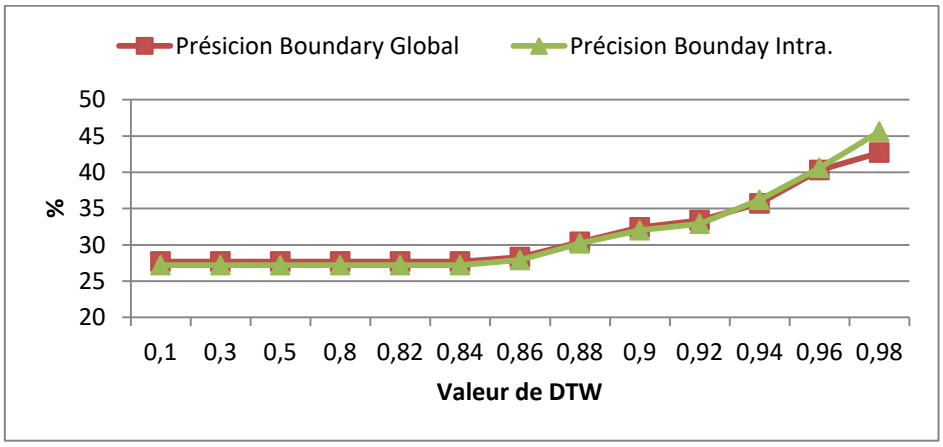


Figure 23 Evolution de la précision de *boundary* en fonction de la valeur de DTW

La précision augmente particulièrement à partir de 0,86. Dans le même temps, le rappel diminue sensiblement pour atteindre 0,2 à une valeur de DTW de 0,98. Ce résultat est attendu dans la mesure où plus la valeur de DTW augmente, moins l'on trouve de segments et par conséquent moins l'on pose de frontière, ce qui explique la diminution du rappel.

On pourrait s'étonner d'obtenir à la fois de faibles résultats pour les *tokens* et des résultats convenables pour *boundary*. Cela est toutefois parfaitement normal. Si l'on prend par exemple le segment trouvé suivant « a bunch of bananas », on peut remarquer que le système a fortement hypo-segmenté : il a trouvé un token là où il devrait y en avoir quatre. Toutefois, il a correctement su poser une frontière avant le token « a » et à après le token « bananas ». De ce fait, il est normal d'avoir un score *token* plus faible que le score *boundary*.

2.2.4 Configuration optimale

Il semblerait que la valeur DTW optimale se situe entre 0,86 et 0,90. Les résultats obtenus à des valeurs supérieures peuvent être meilleurs sur certains points, mais la couverture du corpus chute énormément. C'est à ces valeurs que les scores de précision de *token* et de *type* sont les plus élevées. De plus, à 0,90, le *clustering* semble être de meilleure qualité qu'à d'autres valeurs.

2.3 Comparaison avec les résultats obtenus sur un autre corpus lors de ZeroSpeech15

Afin de pouvoir mettre en perspective nos résultats, nous les comparerons à ceux obtenus lors de la compétition ZeroSpeech15 et présentés dans l'article de (Versteegh et al., 2016).

La *baseline* correspond à ZRTools, le système que nous avons-nous-même utilisé. La *topline* correspond aux résultats d'une grammaire adaptative (comme présentée au point 2.4.1 de l'état de l'art). L'approche adoptée par (Räsänen et al., 2015) est celle que nous avons présentée dans l'état de l'art au point 2.3.2 et procède par découpage du signal en syllabes. Le système de (Lyzinski et al., 2015) utilise une variante de ZRTools. Les expériences ont été menées sur 5 heures de parole en anglais.

English	System	NED	Cov	Type P	Type R	Type F	Token P	Token R	Token F	Boundary P	Boundary R	Boundary F
baseline		**21.9**	16.3	6.2	1.9	2.9	5.5	0.4	8.0	44.1	4.7	8.6
topline		*0.0*	*100.0*	*50.3*	*56.2*	*53.1*	*68.2*	*60.8*	*64.3*	*88.4*	*86.7*	*87.5*
Räsänen et al. [8]	Vseg	89.6	40.6	13.5	11.3	12.3	21.6	4.8	7.9	**76.1**	28.5	41.4
	EnvMin	88.0	42.2	12.7	10.8	11.6	21.6	4.7	7.8	75.7	27.4	40.3
	Osc	70.8	42.4	**14.1**	**12.9**	**13.5**	**22.6**	**6.1**	**9.6**	75.7	33.7	**46.7**
Lyzinski et al. [7]	CC-PLP	77.3	25.5	4.7	2.5	3.3	4.2	0.6	1.0	39.6	7.5	12.7
	CC-FDLPS	61.2	**80.2**	3.1	9.2	4.6	2.4	3.5	2.8	18.8	**64.0**	29.0
	FG-BNF	*36.4*	*46.7*	*2.3*	*2.9*	*2.6*	*1.9*	*0.7*	*1.0*	*31.7*	*14.2*	*19.6*

Figure 24 Résultats ZeroSpeech15 - Extraits de 'Table 2' de l'article (Versteegh et al., 2016)

Afin de pouvoir plus facilement comparer les résultats, nous rappelons nos résultats dans le tableau ci-dessous. Puisque la *baseline*[40], ne donne pas les résultats de *grouping* ni de *matching*, nous ne les avons pas inclus.

Expérience	NED	Couv.	Type P	Type R	Type F	Token P	Token R	Token F	Boundary P	Boundary R	Boundary F
La nôtre (0,86)	24,7	8,9	0,5	0,1	0,2	0,3	0	0	28,3	3,2	5,8
La nôtre (0,90)	17,2	5,9	0,6	0,1	0,2	0,4	0	0	32,4	2,1	4

Tableau 6 Rappel de nos résultats sur le corpus MSCOCO

Nous pouvons constater que les résultats de nos expériences sont moins bons que les résultats de la *baseline*. Les résultats de la *baseline* restent toutefois très faibles et confirme que la segmentation de ZRTools ne permet de faire que très rarement une segmentation au niveau des tokens comme le montre les résultats obtenus pour *type* et *token*. La NED obtenue par la *baseline* reste relativement proche de la nôtre, même si notre couverture est plus faible. Le fait que nous ayons une faible couverture explique notamment que notre score *boundary* soit plus faible. En effet, moins l'on couvre de corpus, moins l'on est à même de poser des frontières.

Le fait que nous ayons obtenus de faibles résultats peut être dû au fait que le corpus est composé de 10.000 éléments de quelques secondes. Avoir des fichiers WAV plus long permettrait probablement d'améliorer les résultats. De nouvelles expériences doivent donc être menées en concaténant l'ensemble des fichiers WAV appartenant à une même image. Cela permettrait ainsi de vérifier notre hypothèse.

On peut également noter que la *topline* a obtenu d'excellents résultats. Toutefois, il faut garder en tête que l'approche prise par la *topline* n'est pas *zero resource* puisque les grammaires adaptatives

[40] (Versteegh et al., 2016) ne mentionnent pas la valeur de DTW utilisée

ont été utilisées sur une version transcrite des fichiers audio. Il convient plus de considérer ce score comme « une référence pour la valeur maximum qu'il est possible d'obtenir avec les métriques utilisées ».[41]

Les résultats obtenus par (Räsänen et al., 2015) sont particulièrement intéressants et montrent comment la segmentation du signal en syllabes permet de limiter les hypo-segmentations comme le révèlent les résultats obtenus pour *token* et *boundary*.

Ces résultats montrent cependant que notre corpus, bien que synthétique, n'est pas irréaliste. Nous aurions pu nous attendre à des résultats bien meilleurs avec un corpus synthétique, ce qui n'est pas le cas. Comme nous l'avons déjà mentionné, nous subodorons que la durée des fichiers WAV a un fort impact sur les résultats.

Le type de corpus utilisé impacte également les résultats. Le corpus utilisé pour le *ZeroSpeech15* est un sous ensemble du *BuckEye Corpus*[42]. Ce sous ensemble du corpus était composé de 12 locuteurs en tout (6 hommes et 6 femmes). Nous pouvons constater que les locuteurs ont fait de nombreuses pauses comme le montre l'extrait suivant :

```
s2403b 32.944 33.084 uh
s2403b 33.084 33.126 SIL
s2403b 33.517 33.864 foreign
s2403b 33.864 34.255 country
s2403b 34.255 34.495 gets
s2403b 34.495 34.565 it
s2403b 34.565 34.594 SIL
s2403b 34.594 34.904 yknow
s2403b 34.904 35.109 and
s2403b 37.261 37.468 well
s2403b 37.468 37.498 uh
```

Figure 25 Extrait du corpus Buckeye[43]

Nous pensons que les nombreux silences, même courts, entre les groupes de mots permet d'aider à segmenter le corpus. De plus, certains extraits sont composés de nombreuses disfluences entre les mots ce qui peut également aider à obtenir une meilleure segmentation.

[41] http://sapience.dec.ens.fr/bootphon/2015/page_5.html [consulté le 24 mai 2017]

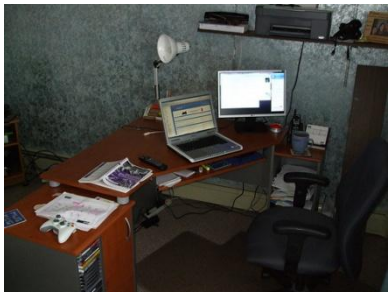
[42] Corpus de parole spontanée disponible à l'adresse suivante http://buckeyecorpus.osu.edu/ [consulté le 24 mai 2017]

[43] Extrait de https://raw.githubusercontent.com/bootphon/tde/master/bin/resources/english.wrd [consulté le 24 mai 2017]

2.4 Présentation de quelques clusters trouvés

A partir de la liste de *clusters* donnée en sortie par ZRTools, nous avons écrit un script Python permettant d'extraire les segments trouvés dans chacun des fichiers WAV et de les concaténer ensemble s'ils appartiennent à un même *cluster*. Grâce à l'API que nous avons développée, nous avons ensuite extrait la transcription correspondant au segment trouvé. Les résultats que nous présentons par la suite sont les résultats que nous avons obtenus pour une valeur de DTW de 0,88. ZRTools a parfois tronqué certains mots que nous avons indiqués en les soulignant.

- **Clusters pertinent vis-à-vis des objets figurés dans l'image**

Numéro du cluster	Images	Fichiers WAV et Transcription
1551	 	134503_71420_Paul_End_0-9 133645_469648_Paul_None_1-0 **Boat <u>with</u>**
1706	 	459934_432216_Elizabeth_None_1-0 197125_33832_Elizabeth_None_1-1 **Computer <u>monitor</u>**

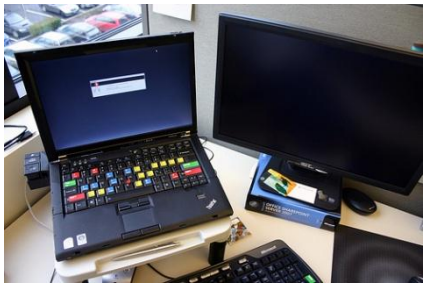
1707	 	459934_432216_Elizabeth_None_1-0 403107_357714_Elizabeth_None_1-0 **Computer <u>monitor</u>**
1711	 	462213_708020_Amanda_None_0-9 416885_129919_Amanda_None_1-0 **<u>A table</u>**
260		459374_466307_Phil_None_0-9 415408_311368_Phil_None_1-0 **Fire hydrant <u>on</u>**

Tableau 7 Exemple de clusters contenant un mot pertinent vis-à-vis des objets dans l'image

- **Expressions polylexicales**

606	 	371683_156894_Judith_None_0-9 1153_459647_Judith_None_0-9 **A bunch of bananas**
1140		218997_136520_Bruce_Beginning_0-9 462565_120854_Bruce_None_1-0 **Group of**

Tableau 8 Exemple de cluster contenant des expressions polylexicales

2.5 Discussion sur nos résultats

2.5.1 Même mots sur différents clusters

(Bansal et al., 2017) ont fait des observations auxquelles nous adhérons totalement. La principale remarque que nous pouvons faire est qu' « un même mot de différents locuteurs apparaît souvent dans de multiples clusters » (Bansal et al., 2017). Nos tests avec une valeur de DTW de 0,88 montrent également que les mots d'un même locuteur sont également séparés dans différents clusters, comme le montre l'exemple suivant :

ID cluster	ID pseudoterme	Nom WAV	Transcription	Frame[44] début	Frame fin
2240	8771	**132982_530830_Bruce_None_1-1**	double decker buses	**10**	**72**
	8772	329614_448261_Phil_None_1-0	double deckered bus	7	75
2241	8773	**132982_530830_Bruce_None_1-1**	double decker buses	**10**	**61**
	8774	394681_121993_Bruce_None_0-9	double decker red	50	110
2242	8775	**132982_530830_Bruce_None_1-1**	double decker buses	**10**	**74**
	8776	263011_429396_Bruce_None_1-1	double decker buses	188	254
2243	8777	**132982_530830_Bruce_None_1-1**	double decker buses	**11**	**67**
	8778	330554_50889_Phil_None_0-9	a blender with	1	65

Tableau 9 Exemple d'un même segment réparti sur plusieurs clusters

Les résultats obtenus par (Lyzinski et al., 2015) lors du *ZeroSpeech15* ont partiellement résolu ce problème. En effet, en changeant les vecteurs d'entrée —des vecteurs autres que le PLP —et en utilisant d'autres méthodes de clustering, on obtient de meilleurs résultats.

[44] https://github.com/arenjansen/ZRTools signale $frame = seconds \times 100$. [consulté le 24 mai 2017]

2.5.2 De la non parole dans les clusters

(Bansal et al., 2016) remarquent que « les silences et les disfluences sont considérés comme des segments valides » notamment car ils « sont fréquents et forment des bonnes paires acoustiques » (Bansal et al., 2016). Nous avons en effet constaté le même phénomène dans notre corpus, où certaines disfluences ont été repérées.

Toutefois, le nombre de disfluences trouvées reste relativement faible et centré sur quelques locuteurs : nous avons par exemple trouvé plus de 80 segments de « uh » chez Jenny mais nous ne l'avons pas identifié chez d'autres locuteurs. La disfluence « um » a été identifiée chez plusieurs locuteurs, bien qu'en nombre bien plus faible que les « uh ».

(Bansal et al., 2016) considèrent dans le cadre de leurs recherches que la découverte de disfluences est du bruit. Ceci n'est pas le cas pour nous car elles sont intéressantes dans le cadre de la documentation des langues. En effet, elles peuvent varier d'une langue à l'autre, il peut donc être intéressant d'avoir des clusters uniquement constitués de disfluences.

2.5.3 Impureté des clusters

Certains clusters sont impurs car « des paires phonétiquement similaires identifiées par le système sont sémantiquement différentes » (Bansal et al., 2016). Nous amenderons cette citation et dirons que des paires *acoustiquement* similaires ne garantissent pas qu'elles soient phonétiquement similaires et par conséquent qu'elles soient sémantiquement proches. En effet, deux sons peuvent être acoustiquement proches mais former des phonèmes différents dans la langue en question, et par conséquent peuvent impliquer des changements sémantiques. Cela est particulièrement vrai dans le cas des langues tonales.

Ce phénomène s'est vérifié dans notre corpus, où des segments ont été identifiés comme similaires alors qu'ils sont différents :

ID cluster	WAV	Transcription
17	416827_69202_Paul_End_0-9	Through **a mon**tainous
	131280_586577_Paul_None_0-9	To **a man**
1701	374485_494269_Amanda_None_1-1	An **old** van
	395283_708183_Amanda_None_0-9	Man **hold**ing a

Tableau 10 Exemple de clusters impurs résultant d'une grande ressemblance acoustique

Toutefois, un tel phénomène peut également être considéré comme une force dans le cas des langues où il existe des mutations consonantiques, soient à l'intérieur ou à l'initiale des mots, comme

c'est le cas pour les langues celtiques. Un tel système permettrait possiblement de repérer « vras » et « bras » ou bien « mawr » et « fawr » comme étant un même mot.

Breton	Gallois	Significaiton
gwreg	gwraig	*femme*
bras	mawr	*Grand*
ar **w**reg **v**ras	y **w**raig **f**awr	*La grande femme*

Tableau 11 Exemple de mutation consonantique en Breton et en Gallois[45]

Ainsi, la notion de pureté des clusters et de bruit dépend largement de la langue étudiée et de l'utilisation que l'on fait des clusters automatiquement découverts.

2.5.4 Lien entre les images et les segments audio

SI nous nous mettons dans le contexte d'un linguiste qui aurait à documenter une langue, plusieurs solutions s'offrent à lui pour savoir à quoi correspond le segment audio identifié dans plusieurs images.

La première et la plus évidente est si l'une des images ne figure qu'un seul objet et ce de façon iconique. C'est par exemple le cas du cluster 260 (visible dans la partie 2.4 de ce présent chapitre) où les deux images montrent une bouche d'incendie de manière iconique. C'est également le cas du cluster 1707 où l'une des images figure des moniteurs d'ordinateur de façon iconique.

Toutefois, même si les images ne figurent pas d'objets iconiques, il peut être relativement aisé de savoir à quel objet ou action se rapporte l'objet s'ils sont dans des contextes très différents. C'est ainsi le cas du cluster 1706 où le seul point commun entre les deux images est un ordinateur.

Le cluster 1551 est spécial puisque les images présentent plusieurs éléments en commun : un bateau et de l'eau. Dans de tels cas, le linguiste pourra circonscrire la signification du segment à l'un de ces deux éléments, sans toutefois pouvoir privilégier l'une ou l'autre des interprétations. Afin de pouvoir lever le doute, il sera nécessaire de croiser ces informations avec d'autres clusters où les images figurent l'un ou l'autre des éléments.

[45] Le tableau provient de la page suivante https://en.wikipedia.org/wiki/Consonant_mutation#Celtic_languages [consulté le 24 mai 2017]

2.5.5 Hypo-segmentation

(Bansal et al., 2017) remarquent que ZRTools « est plus fiable sur des motifs longs et fréquemment répétés. ». Ce phénomène s'est également vérifié lors de nos expériences. Nous avons constaté que la segmentation faite par ZRTools est en général faite au niveau des chunks. Il serait à ce titre intéressant d'ajouter une mesure à celles proposées par (Ludusan et al., 2014) permettant de mesurer à quel point les chunks trouvés par ZRTools correspondent à des chunks réels.

Ce phénomène d'hypo-segmentation est souvent considéré comme un problème, puisque l'on souhaite en général que la segmentation corresponde aux *gold tokens*. Le fait de reconnaître des chunks entiers comme étant un seul mot semble toutefois être un phénomène tout à fait normal. En effet, (Bannard and Matthews, 2008) signalent que les enfants ont tendance à stocker en mémoire des séquences entières de mots correspondant à des chunks et ayant une fréquence d'apparition élevée. Ces mêmes auteurs signalent également que les enfants ont différents niveaux de granularité, allant « des mots, aux bigrammes, etc. ». Ces différents niveaux de granularité dans la segmentation sont aussi apparus sur notre corpus :

Numéro du chunk	Transcription des segments[46]	Fréquence d'apparition dans notre corpus de test Rang d'apparition
Unigrammes		
197	standing	0,63% 17e/4.668
141	uh	0,51% 21e/4.668
Bigrammes		
19	Sitting on	0,43% 10e/31.716
887	A woman	0,38% 13e/31.716
Trigrammes		
2 3 4	The living <u>room</u>	0,007% 1018e/66.153
27	Sitting on a	0,23% 5e/66.153
Quadrigrammes		
1168	The middle of a	0,03% 27e/89.691
1171	In the middle of	0,05% 13e/89.691
Pentagrammes		

[46] Les mots soulignés indiquent qu'ils ont été tronqués

167	A man and a woman	0,02% 11e /100.912
1085	Player is getting ready to hit	% 1152e/100.912

Tableau 12 Exemple de n-grammes fréquent trouvés par ZRTools

3 Expérimentations sur le corpus Mboshi

Nous avons également fait des expériences sur le corpus en Mboshi. Celui-ci n'est constitué que de 41 minutes de parole, ce qui est très peu. Cela nous permettra de voir si l'algorithme est performant sur un jeu de données relativement réduit mais réaliste du point de vue de la documentation d'une langue inconnue.

Nous avons fait plusieurs expérimentations : soit en utilisant les fichiers audio dans leur ensemble soit en utilisant une version découpée des mêmes fichiers audio. Le découpage a été fait manuellement lorsqu'il y avait un silence – afin d'éviter de découper un mot. Les fichiers ont été découpés en segments d'une vingtaine de secondes environ. En effet, nous avons pensé qu'en pré-segmentant les fichiers sur les silences, le système pourrait plus facilement distinguer des segments communs. Nous présenterons les résultats que nous avons obtenus avec la version découpée du corpus puisque c'est avec cette méthode que nous avons obtenus les meilleurs résultats.

Les résultats sur notre corpus synthétique en anglais ont montré que la valeur optimale de DTW était située entre 0,86 et et 0,90. Nous avons donc choisi d'utiliser une valeur de DTW de 0,89 pour nos expérimentations sur le corpus en Mboshi.

3.1 Métrique d'évaluation

Puisque nous ne disposons d'aucune transcription pour le corpus Mboshi obtenu par élicitation, la boîte à outils de (Ludusan et al., 2015) ne peut être utilisée. L'évaluation des résultats a dû être faite manuellement. Dans le cas du Mboshi, il s'agit de savoir si le système a su extraire le mot désignant l'objet figuré dans l'image. Ainsi, pour évaluer nos résultats, nous avons regardé si le mot désignant l'objet figuré dans l'image était présent dans au moins l'un des clusters associés à l'image. Si tel est était le cas, nous considérions le mot comme trouvé.

Le corpus est constitué de 30 images, donc théoriquement nous devrions retrouver 30 mots différents. Toutefois, comme mentionné au point 1.1 du chapitre 2, sur les 30 images, nous n'avons nous même pas retrouvé les 30 mots différents, mais seulement 28. Ainsi, pour la suite, nous allons considérer que le nombre maximal de mots que pouvait retrouver le système était de 28.

3.2 Expérimentations

Lors de nos tests sur le corpus MSCOCO, nous avions laissé les paramètres par défaut. Pour nos expérimentations sur le Mboshi, nous avons modifié les paramètres de création des clusters afin de voir l'influence que ceux-ci avaient sur le résultat final.

3.2.1 Modification des paramètres

Nous avons modifié deux paramètres :

- DURTHR (seuil de durée) : permet de prendre en compte les segments ayant une durée supérieure ou égale à la durée spécifiée. Cette valeur est fixée par défaut à 50 trames, soit une demie seconde. Nous avons abaissé ce seuil à 0,30 seconde. Cela permet de prendre en compte les segments plus courts.

- OLAPTHR (seuil de chevauchement) : ce paramètre permet de résoudre — en partie — le problème que nous avons souligné au point 2.5.1 de ce présent chapitre où un segment appartenant à un même locuteur est séparé sur différents clusters. Ce paramètre permet de réunir les clusters si ceux-ci comportent des segments qui se chevauchent. Par défaut, ce paramètre est fixé à 0,97 et nous l'avons réduit à 0,10. Ainsi, il suffit que deux segments se chevauchent de 10% pour qu'ils soient réunis. Changer ce paramètre entrainera inévitablement la présence de bruit dans certains clusters, toutefois, cela permettra que les clusters contenant le mot pertinent soient réunis et ainsi qu'ils soient plus visibles.

3.2.2 Résultats

Les Tableau 13 présente les résultats obtenus sur la version découpée du corpus. Nous pouvons constater que ZRTools a réussi à trouver 9 mots. Le système a produit en tout 105 clusters.

Nom du fichier WAV	Mot à retrouver	Mot retrouvé	Clusters pertinents	N° cluster	Nombres de segments appartenant au fichier dans le cluster	Nombre de segments dans le cluster	Rappel	Précision
[…]_Elicit_0.wav	Alerere	Alerere	2	25	3	3	0,45	1
				100	2	2		
[…]_Elicit_13.wav	Bvue a ibyee	Bue a ibyee	1	35	2	2	0,5	1
[…]_Elicit_15.wav	dongodongo	dongodongo	1	47	5	5	0,83	1
[…]_Elicit_16.wav	dumuledzooni	dumuledzooni	1	92	4	4	1	1
[…]_Elicit_23.wav	Edzesa	Edzesa	1	9	2	2	0,18	1
[…]_Elicit_28.wav	Esondo ya ondongo	Ondongo onlongo	1	22	8	8	0,72	1
[…]_Elicit_3.wav	Aswebhe	Aswebhe	1	76	2	2	0,3	1
[…]_Elicit_5.wav	Atsinga	Atsinga	1	28	1	2	0,05	0,5
[…]_Elicit_7.wav	Ayaa	Ayaa	1	44	1	14	0,25	0,57

Tableau 13 Résultats obtenus avec la version découpée du corpus

Lorsque le numéro des clusters est souligné, cela signifie que le système a tronqué une partie du mot. Toutefois, même si celui-ci est tronqué, il reste reconnaissable.

Nous pouvons constater que dans l'ensemble notre précision est assez élevée ce qui montre qu'il y a peu de bruit dans les clusters. Le cluster pour le mot « ayaa » n'a été retrouvé qu'une seule fois dans le fichier WAV « elicit_7 », toutefois ce mot est également utilisé dans d'autres fichiers WAV, ce qui explique que sa précision soit assez élevée.

La modification des paramètres a permis que les occurrences d'un même mot soient réunies au sein d'un même cluster, ce qui n'est pas le cas avec les paramètres par défaut comme nous l'avons déjà fait remarquer avec l'anglais. Ainsi, nous avons retrouvé 9 mots sur les 28 qui pouvaient être retrouvés, soit 32% des mots.

3.3 Repérer les segments pertinents et exclure les segments communs

Dans le cas d'un corpus comme celui que nous avons en Mboshi, nous ne pouvons pas utiliser les mêmes techniques qu'avec le corpus en anglais pour déterminer la signification des segments. En effet, chaque objet n'est figuré qu'une fois et n'apparaît pas dans d'autres images. Il est donc nécessaire de mettre au point une nouvelle méthode afin d'essayer de trouver le nom de l'objet. Il s'agit donc de trouver pour chacune des images le cluster le plus représentatif et d'exclure les clusters les moins représentatifs.

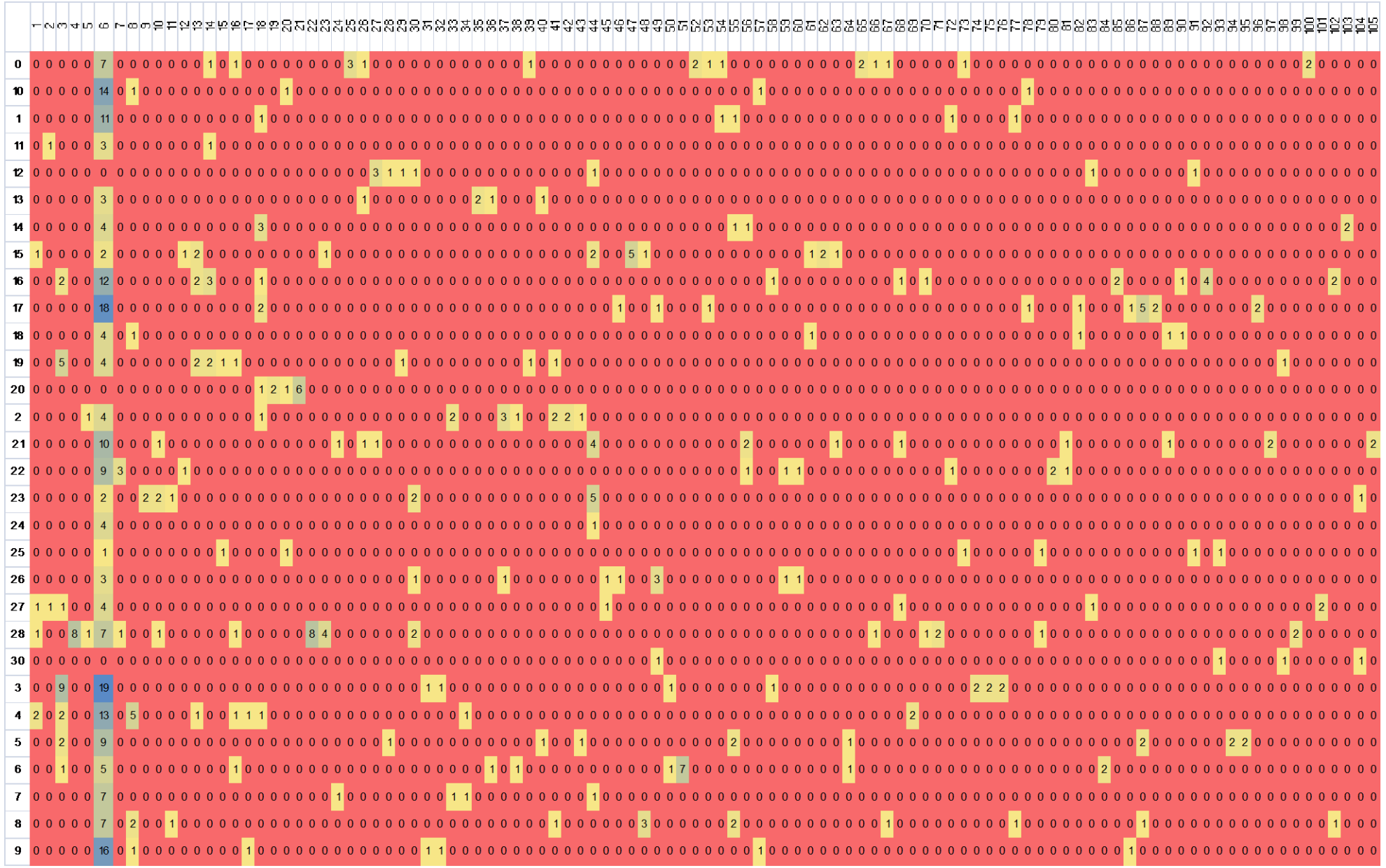
Dans le Tableau 14, nous présentons les résultats que nous avons obtenus sur la version découpée du corpus en Mboshi. La première colonne indique le numéro des fichiers WAV. La première ligne indique le numéro des différents clusters trouvés. L'intersection d'une ligne et d'une colonne indique le nombre de segments appartenant au fichier WAV qui ont été placés dans le cluster.

On sait que chacun des objets n'est figuré que sur une seule image. Ainsi, le mot le désignant devrait théoriquement apparaître plusieurs fois dans un seul fichier WAV, celui associé à l'image, et pas dans les autres fichiers WAV. Ainsi, si l'on constate qu'un cluster inclut des segments appartenant à de nombreux fichiers WAV, les segments qu'il contient ne sont pas pertinents vis-à-vis des objets figurés dans les images associées aux fichiers WAV. De tels clusters sont visibles sur la matrice car ils forment une colonne, comme par exemple les clusters suivant : 3, 6, 8, 14, 18, 30 et 44.

Au contraire, si un cluster ne contient que des segments provenant d'un seul fichier WAV, c'est que ceux si sont très pertinents vis-à-vis du fichier WAV et donc de l'objet figuré dans l'image. C'est par exemple le cas des clusters n°25, 35, 47, 92, 9 et 22.

Tableau 14 Représentation des résultats obtenus pour le Mboshi sous forme de matrice

La première colonne indique le numéro des fichiers WAV. La première ligne indique le numéro des différents clusters trouvés. L'intersection d'une ligne et d'une colonne indique le nombre de segments appartenant au fichier WAV qui ont été placés dans le cluster.



Toutefois, cette matrice n'est que très difficilement lisible, surtout si elle contient un grand nombre de fichiers WAV et un nombre important de clusters. Il convient donc d'automatiser la recherche du cluster le plus pertinent pour une image.

Pour ce faire nous avons utilisé une mesure de TF-IDF. Normalement, cette mesure est faite sur des textes et permet de déterminer le token ou le n-gramme le plus pertinent pour un document à l'échelle de tout un corpus. Nous nous en sommes servi ici pour déterminer quel est le cluster le plus pertinent pour une image vis-à-vis de son apparition dans le reste du corpus.

Les mesures TF-IDF nécessitent un texte en entrée. Il nous a donc fallu représenter le contenu de chaque fichier WAV de manière textuelle. Pour ce faire, nous avons concaténé le numéro des clusters qui contenaient un segment appartenant au fichier WAV, et avons répété ce numéro autant de fois qu'il y avait de segments appartenant au fichier WAV dans le cluster. Voici par exemple la représentation textuelle du fichier WAV n°0 : « 6 6 6 6 6 6 6 14 16 25 25 25 26 39 52 52 53 54 65 65 66 67 73 100 100 »

En faisant cela pour chacun des fichiers WAV nous avons pu nous servir des mesures TF-IDF pour déterminer les clusters les plus pertinents et les moins pertinents pour une image[47]. Les résultats sont visibles en annexe à la page 139. Nous pouvons constater que sur les 9 mots trouvés, 5 obtiennent les scores TF-IDF les plus élevés. Nous pouvons également voir que les clusters que nous avions identifiés comme non pertinents grâce à la matrice obtiennent en général avec les scores TF-IDF les plus faibles. Cependant, le cluster n°44 que nous avions suspecté être non pertinent, s'avère contenir le mot « ayaa » qui correspond à un mot que nous recherchions. Cela s'explique par le fait qu'il est répété dans d'autres fichiers WAV que celui associé à son image.

Afin de vérifier si les clusters que nous avons identifiés comme non pertinents le sont vraiment, nous avons demandé à un locuteur du Mboshi – Guy-Noël Kouarata – leur signification :

Numéro du cluster	Signification	Remarques
3	ayɛli ɛ « pas moyen »	
6	ɛɛ, ware waa « oui, il parait que si.... »	Le cluster contient également des disfluences comme « eh »
14	lebvulu obvula « on est encore revenu »	

[47] Nous avons adapté le code de la page suivante : http://stevenloria.com/finding-important-words-in-a-document-using-tf-idf/ [consulté le 24 mai 2017]

18	nga liitaa mia « moi je les vois »	
30	nzaa ba abe « mais les mauvais »	

Tableau 15 Signification des segments contenus dans les clusters supposés non pertinents vis-à-vis des objets figurés dans les images

Les traductions fournies par Guy-Noël Kouarata confirment donc notre hypothèse. Les clusters formant une ligne sur la matrice et ayant les scores TF-IDF les plus faibles sont donc non pertinents vis-à-vis des images décrites. Ainsi, le score TF-IDF sur les clusters semble être une mesure pertinente pour notre corpus afin de distinguer les clusters pertinents pour l'objet figuré dans l'image des clusters non pertinents.

Toutefois, il faut relativiser nos résultats car notre corpus est très petit. Dans le cas d'un corpus plus grand, il est fort probable que beaucoup de mots auraient eu une répartition similaire à celle du mot « ayaa » qui est apparu dans la description d'autres images et non pas uniquement sur celle dont il était l'objet principal. Ainsi, même si ces résultats sont encourageants, ils restent à confirmer sur un corpus d'une plus grande taille.

4 Conclusion

Dans ce chapitre, nous avons décrit l'algorithme d'un système de découverte automatique de formes similaires à partir du signal, ZRTools, et l'avons appliqué à la détection non supervisée de mots sur deux corpus multimodaux. Nous avons montré qu'un tel **système** pouvait être **utilisé pour découvrir de manière non supervisée du lexique** et permettre ainsi de **commencer à documenter une langue en danger.**

Conclusion et perspectives

1 Conclusion

Nous nous sommes intéressé dans notre travail de recherche à la découverte non supervisée de lexique appliquée à la documentation des langues en danger. **La contribution de nos travaux à la communauté scientifique s'est faite à trois niveaux :**

1. **Nous avons ajouté une nouvelle modalité — la parole — à un corpus multimodal déjà existant : MSCOCO.** L'ajout d'une telle modalité offre de **nouvelles perspectives d'utilisation** à ce corpus et permet ainsi qu'il soit utilisé pour faire de la découverte non supervisée de lexique (*spoken term discovery* ou bien encore *keyword spotting*), pour modéliser l'acquisition du langage dans un contexte multimodal, ou bien encore pour la projection d'images et de parole dans des espaces de représentation communs à l'image des travaux déjà entrepris par (Harwath and Glass, 2015) et (Kamper et al., 2017).

   Le corpus que nous avons créé dispose de nombreux atouts, notamment le fait de pouvoir connaitre avec précision les étiquettes temporelles (*timecodes*) de chacun des mots, syllabes et phonèmes. De plus, nous avons créé une API permettant de pouvoir facilement le manipuler et de trouver les légendes selon des critères fournis par l'utilisateur.

   Notre corpus sera librement accessible afin qu'il puisse être largement utilisé par les communautés scientifiques sus-mentionnées**. Il sera d'ailleurs utilisé dans le cadre du *Jelinek Memorial Workshop on Speech and Language Technology 2017*** qui se tiendra pendant 6 semaines à l'université Carnegie Mellon (Pennsylvanie – USA) et dont le thème est le suivant : *The Speaking Rosetta Stone - Discovering Grounded Linguistic Units for Languages without Orthography*.

   Le corpus que nous avons créé pourrait encore être amélioré, en travaillant plus finement la prosodie des phrases produites. Il serait également possible de travailler l'emphase de certains mots, afin que notamment les mots clefs des images ressortent plus.

2. Nous nous sommes servis de notre corpus synthétique afin de faire de la **découverte non supervisée de lexique dans un contexte multimodal**. Nous avons montré que même si notre corpus est synthétique il n'en reste pas moins réaliste au vu de la variabilité intra- et interlocuteurs et des disfluences qu'il contient.

Celui-ci nous a également permis d'évaluer les performances d'un système de découverte non supervisée de lexique et permis de **dégager les paramètres optimums**. Nos tests nous ont également permis de voir les limites du système utilisé et d'entrevoir des améliorations éventuelles pour des travaux futurs.

Nos tests nous ont également permis de mettre au point différentes méthodes afin de relier les segments découverts aux images et ainsi connaitre leur signification.

3. Finalement, notre dernière contribution à la communauté scientifique **est l'application d'un système de découverte non supervisée de lexique dans un contexte multimodal et son évaluation sur une vraie langue en danger : le Mboshi.**

   Nous avons montré qu'il est possible de découvrir automatiquement du lexique de manière totalement non supervisée, et ce pour une langue pour laquelle nous ne disposions d'aucune information au départ. Nous avons également mis au point une méthode, inspirée de la recherche d'information, afin de repérer les clusters qui étaient pertinents vis-à-vis des objets figurés dans les images et avons pu exclure ceux qui ne l'étaient pas.

   Nous avons également pu constater que de nombreux mots n'ont pas été trouvés alors qu'ils avaient un nombre d'occurrences élevé. Cela montre qu'il reste du travail à faire afin d'améliorer la découverte non supervisée de lexique pour les langues non écrites. La modification d'autres paramètres reste à explorer afin de voir s'ils permettent d'améliorer les résultats.

2 Perspectives

Les méthodes et logiciels que nous avons employés ne peuvent être mis en œuvre que par un utilisateur ayant des compétences en linguistique informatique, ce qui n'est pas forcément le cas des linguistes de terrain. ZRTools notamment peut être d'un abord compliqué puisqu'il nécessite de grandes ressources pour fonctionner et qu'il est utilisable uniquement en ligne de commande. De la même manière, nos résultats TF-IDF peuvent être difficilement compréhensibles pour une personne ne maitrisant pas cette mesure. Ainsi, il serait nécessaire de prévoir une **interface permettant d'utiliser facilement ces outils** et de rendre les résultats aisément interprétables.

Afin d'améliorer le corpus que nous avons conçu nous pensons ajouter une nouvelle modalité. Nous songeons en effet à **traduire automatiquement en Japonais chacune des légendes de MSCOCO**. L'ajout

de traduction permettrait notamment de travailler sur la projection des images, de la parole et des traductions dans un espace de représentation commun. L'aide apportée par la traduction permettrait certainement d'améliorer la découverte du lexique comme le montre les travaux de (Bansal et al., 2016).

Nos tests de traduction en Japonais[48] permettent d'obtenir des traductions, qui sans être parfaites, permettent de retranscrire l'idée principale de la phrase. Ces traductions permettraient de mesurer la capacité d'un système non supervisé à découvrir du lexique avec l'aide d'informations bruitées, comme cela peut parfois être le cas avec des données réelles.

Le corpus en Tima dispose également de transcriptions, de traductions ainsi que de listes de mots enregistrées par différents locuteurs. De nombreux travaux peuvent être menés afin de voir si ces informations supplémentaires permettent d'obtenir de meilleurs résultats tout en restant dans un contexte semi supervisé (utilisation d'une petite quantité d'information expertes).

Nous aurions également souhaité savoir si **l'utilisation d'un système de vision par ordinateur** permettrait d'émettre des hypothèses de signification aux segments trouvés, à la lumière du contenu des images. Cela peut être compliqué dans le cas de certains corpus comme pour le Mboshi, où les images sont très liées au contexte immédiat et figurent des plantes et animaux endémiques. Les systèmes de vision par ordinateur ne seraient pas capables d'identifier correctement le contenu des images. Toutefois, dans le cas des corpus en Tima et en Tabaq, les images figurent parfois des objets communs – vaches, chèvres, vélo, personnes – que des systèmes de vision par ordinateur parviendraient à reconnaitre. Les sorties de tels systèmes, couplés à des scores TF-IDF permettraient sûrement de trouver la signification de certains segments.

Nous souhaiterions également étudier **l'influence de la prosodie** et voir à quel point celle-ci affecte les résultats. Comme nous l'avons déjà mentionné, de nombreux mots ayant un nombre d'occurrences élevé n'ont pas été trouvés, et cela est probablement dû au fait que la prosodie adoptée lors de leur prononciation était trop différente. De manière générale également, il est rare de trouver, dans un même cluster, des mots appartenant à différents locuteurs. Ceci pose la question de la **robustesse à la variabilité interlocuteur** des approches « par l'exemple » travaillant directement sur le signal de parole, comme ZRTools. Les travaux de (Lyzinski et al., 2015) ont notamment montré que les vecteurs d'entrées utilisés semblent avoir une forte influence sur les résultats.

[48] En utilisant le site http://www.excite.co.jp/world/english_japanese/ [consulté le 24 mai 2017]

Comme nous l'avons mentionné, ZRTools a tendance à hyposegmenter et à considérer des chunks entiers comme étant un seul token. Nous souhaiterions savoir si ZRTools peut procéder à une segmentation plus fine, en redonnant par exemple les segments découverts comme nouvelle entrée. Une segmentation plus fine permettrait notamment **d'accéder à des éléments de morphologie et de syntaxe.**

Ainsi, si nous prenons le cluster du Tableau 16, pourrait-on travailler sur la morphologie et voir que l'on a deux mots formés sur le même modèle : radical + *-ing* ?

Numéro du cluster (corpus MSCOCO)	Transcription des segments
2465	Skateboard rid**ing** on a Broccoli lay**ing** on

Tableau 16 Exemple de clusters permettant de travailler sur de la morphologie

Le Tableau 17 montre que l'on pourrait également travailler sur la syntaxe. Nous pouvons constater que l'ensemble des segments formant les clusters contiennent le mot *standing*. Nous pouvons également remarquer que ce même mot est suivi d'un certain nombre de mots différents : *in, on, by, near*. Ces derniers n'apparaissent pas qu'une seule fois derrière *standing* mais de nombreuses fois.

Numéro du cluster (corpus MSCOCO)	Transcription des segments
673	standing **on** the standing **on** a beach standing **on** a standing **on** a
303	standing **in** a standing **in** the standing **in** a room
112	… standing **by** standing **near** …

Tableau 17 Exemple de clusters permettant de travailler sur de la syntaxe

Ainsi, serait-il possible d'envisager de faire de la syntaxe et de regrouper les segments selon leur catégorie morphosyntaxique supposée ? Nous aurions, pour les exemples précédents, un cluster qui contiendrait les mots *standing* et *laying* et un autre contenant les mots *in, on, by,* et *near.* Nous pensons que cet objectif est réalisable et modéliser une proto-syntaxe permettrait ainsi de clairement passer de la *documentation* d'une langue à sa *description*.

Finalement, nous nous demandons s'il serait envisageable de faire de la sémantique si nous avions eu une segmentation plus fine. Les approches du type *word2vec* nous semble être pertinentes afin de réunir des segments qui sont très liés d'un point de vue sémantique – par la synonymie, l'hyperonymie, l'hypomimie, la méronymie par exemple.

Nous avons mentionné que selon (Bannard and Matthews, 2008) les enfants ont tendance à mémoriser des chunks entiers (n-grammes fréquents) et à les considérer comme une seule unité. Serait-il possible, sans faire de segmentation plus fine, d'établir une proto-syntaxe uniquement à partir de ces chunks ?

Nous avons appliqué le système de à la documentation des langues en danger dans un contexte multimodal. Cela a été l'occasion **découverte non supervisée de lexique** de faire un **parallèle avec l'acquisition du langage chez les enfants.** Ainsi, le travail que nous avons mené s'inscrit dans un **contexte plus large** que celui de la documentation des langues en danger. Nous intéresser aux modèles cognitifs existants modélisant l'apprentissage du langage chez les enfants apporterait ainsi de nouvelles perspectives à notre travail. Etudier l'acquisition du langage nous permettrait de confronter les différentes méthodologies employées, tant celles issues de l'informatique, que celles issues de la linguistique et des sciences cognitives.

Nous pourrions donc résumer les perspectives de notre travail en quelques questions :

- Des informations supplémentaires (traductions, listes de mots, vision par ordinateur, etc.) permettraient-elles d'améliorer la découverte non supervisée de lexique ? Lesquelles et dans quelles mesures ?

- Quelle est l'influence des phénomènes suprasegmentaux (accent, rythme, motifs intonatifs, durée des phonèmes, hauteur de la voix, etc.) sur la découverte du lexique ? Ces phénomènes peuvent-ils servir à découvrir plus facilement du lexique ? Doit-on ensuite chercher à limiter les variations qu'ils créent pour avoir des clusters plus uniformes ?

- Est-il possible d'atteindre un niveau plus fin de segmentation ce qui permettrait ainsi de faire de la syntaxe et de la morphologie ?

- Dans quelles mesures le système de découverte non supervisée de lexique que nous avons employé permettrait-il de modéliser l'acquisition du langage ?

Bibliographie

Abdellah Fourtassi, Benjamin Börschinger, Mark Johnson, Emmanuel Dupoux, 2013. Whyisenglishsoeasytosegment? doi:10.13140/2.1.4775.1049

Adda, G., Stüker, S., Adda-Decker, M., Ambouroue, O., Besacier, L., Blachon, D., Bonneau-Maynard, H., Godard, P., Hamlaoui, F., Idiatov, D., Kouarata, G.-N., Lamel, L., Makasso, E.-M., Rialland, A., Van de Velde, M., Yvon, F., Zerbian, S., 2016. Breaking the Unwritten Language Barrier: The BULB Project. Procedia Comput. Sci., SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia 81, 8–14. doi:10.1016/j.procs.2016.04.023

Akinlabi, A., Connell, B., 2008. The interaction of linguistic theory, linguistic description and linguistic documentation., in: Ozo-mekuri, N., Imelda, U., Obonna, A. (Eds.), Critical Issues in the Study of Linguistics, Languages and Literatures in Nigeria: A Festschrift for Conrad Max Benedict Brann. pp. 571–589.

Austin, P., 2010. Current issues in language documentation, in: Language Documentation and Description. SOAS, Londres, pp. 12–33.

Austin, P., Sallabank, J., Grenoble, L., Grinevald, C., Bert, M., Bradley, D., O'Shannessy, C., Palosaari, N., Campbell, L., Michael, L., Spolsky, B., Woodbury, A., Berson, J., Good, J., Conathan, L., Nathan, D., Hinton, L., Lüpke, F., Mosel, U., McCarty, T., Coronel-Molina, S., Holton, G., Harbert, W., Jukes, A., Moriarty, M., Bowern, C., 2011. The Cambridge Handbook of Endangered Languages, 1st ed, Cambridge Handbooks in Language and Linguistics. Cambridge University Press, Cambridge.

Bannard, C., Matthews, D., 2008. Stored word sequences in language learning: the effect of familiarity on children's repetition of four-word combinations. Psychol. Sci. 19, 241–248. doi:10.1111/j.1467-9280.2008.02075.x

Bansal, S., Kamper, H., Goldwater, S., Lopez, A., 2016. Weakly supervised spoken term discovery using cross-lingual side information. ArXiv160906530 Cs.

Bansal, S., Kamper, H., Lopez, A., Goldwater, S., 2017. Towards speech-to-text translation without speech recognition. ArXiv170203856 Cs.

Bérard, A., Pietquin, O., Besacier, L., Servan, C., 2016. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. Presented at the NIPS Workshop on end-to-end learning for speech and audio processing.

Berment, V., 2004. Méthodes pour informatiser les langues et les groupes de langues « peu dotées » (phdthesis). Université Joseph-Fourier - Grenoble I, Grenoble.

Besacier, L., Barnard, E., Karpov, A., Schultz, T., 2014. Automatic speech recognition for under-resourced languages: A survey. Speech Commun. 56, 85–100. doi:10.1016/j.specom.2013.07.008

Besacier, L., Zhou, B., Gao, Y., 2006. Towards speech translation of non written languages, in: 2006 IEEE Spoken Language Technology Workshop. Presented at the 2006 IEEE Spoken Language Technology Workshop, pp. 222–225. doi:10.1109/SLT.2006.326795

Birch, B., 2013. Ma Iwaidja Dictionary [WWW Document]. URL https://play.google.com/store/apps/details?id=com.pollen.maiwaidjadictionary&hl=en (accessed 1.11.17).

Bird, S., Hanke, F.R., Adams, O., Lee, H., 2014. Aikuma: A Mobile App for Collaborative Language Documentation, in: ResearchGate. Presented at the Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages, pp. 1–5. doi:10.3115/v1/W14-2201

Blachon, D., Gauthier, E., Besacier, L., Kouarata, G.-N., Adda-Decker, M., Rialland, A., 2016. Parallel Speech Collection for Under-resourced Language Studies Using the Lig-Aikuma Mobile Device App. Procedia Comput. Sci., SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia 81, 61–66. doi:10.1016/j.procs.2016.04.030

Bortfeld, H., Leon, S., Bloom, J., Schober, M., Brennan, S., 2001. Disfluency Rates in Conversation: Effects of Age, Relationship, Topic, Role, and Gender [WWW Document].

Bowern, C., 2008. Linguistic fieldwork: a practical guide. Palgrave Macmillan, Houndmills, Basingstoke, Hampshire [England] ; New York.

Brenzinger, M., Dwyer, A., de Graaf, T., Grindevald, C., Krauss, M., Miyaoka, O., Ostler, N., Sakiyama, O., Villalón, M., Yamamoto, A., Zepeda, O., 2003. VITALITE ET DISPARITION DES LANGUES : Groupe d'experts spécial de l'UNESCO sur les langues en danger.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollar, P., Zitnick, C.L., 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. ArXiv150400325 Cs.

Chrupała, G., Gelderloos, L., Alishahi, A., 2017. Representations of language in a model of visually grounded speech signal. ArXiv170201991 Cs.

Clark, H.H., Fox Tree, J.E., 2002. Using uh and um in spontaneous speaking. Cognition 84, 73–111.

Cohen, M., 1997. Lettre à Antonio Saura. Pensée Midi 23–33.

Cohen, S., 2016. Bayesian analysis in natural language processing.

Constant, M., Dister, A., 2012. Les disfluences dans les mots composés. Presented at the Journées sur l'Analyse des Données Textuelles, Liège.

Drude, S., Birch, B., Broeder, D., Withers, P., Wittenburg, P., 2013. Crowd-sourcing and apps in the field of linguistics: Potentials and challenges of the coming technology.

Duong, L., Anasosopoulos, A., Chiang, D., Bird, S., Cohn, T., 2016. An Attentional Model for Speech Translation Without Transcription. Proc. NAACL-HLT.

Gelderloss, L., Chrupala, G., 2016. From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning. Proc. COLING 2016 26th Int. Conf. Comput. Linguist.

Gippert, J., Himmelmann, N.P., Mosel, U. (Eds.), 2006. Essentials of language documentation, Trends in linguistics Studies and monographs. Mouton de Gruyter, Berlin.

Godard, P., Adda, G., Adda-Decker, M., Allauzen, A., Besacier, L., Bonneau-Maynard, H., Kouarata, G.-N., Löser, K., Rialland, A., Yvon, F., 2016. Preliminary Experiments on Unsupervised Word Discovery in Mboshi. pp. 3539–3543. doi:10.21437/Interspeech.2016-886

Goldwater, S., Griffiths, T.L., Johnson, M., 2011. Producing Power-Law Distributions and Damping Word Frequencies with Two-Stage Language Models. J Mach Learn Res 12, 2335–2382.

Goldwater, S.J., 2006. Nonparametric Bayesian Models of Lexical Acquisition.

Harwath, D., Glass, J., 2015. Deep Multimodal Semantic Embeddings for Speech and Images. ArXiv151103690 Cs.

Harwath, D., Torralba, A., Glass, J., 2016. Unsupervised Learning of Spoken Language with Visual Context, in: Advances in Neural Information Processing Systems. pp. 1858–1866.

Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. 87, 1738–1752. doi:10.1121/1.399423

Himmelmann, N.P., 2012. Linguistic Data Types and the Interface between Language Documentation and Description.

Jansen, A., Church, K., Hermansky, H., 2010. Towards spoken term discovery at scale with zero resources., in: INTERSPEECH 2010. Makuhari, Chiba, Japan.

Jansen, A., Durme, B.V., 2011. Efficient spoken term discovery using randomized algorithms, in: 2011 IEEE Workshop on Automatic Speech Recognition Understanding. Presented at the 2011 IEEE

Workshop on Automatic Speech Recognition Understanding, pp. 401–406. doi:10.1109/ASRU.2011.6163965

Johnson, M., 2009. Learning rules with Adaptor Grammars (the Google edition).

Johnson, M., 2008a. Unsupervised Word Segmentation for Sesotho Using Adaptor Grammars, in: Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology, SigMorPhon '08. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 20–27.

Johnson, M., 2008b. Using Adaptor Grammars to Identify Synergies in the Unsupervised Acquisition of Linguistic Structure, in: Proceedings of ACL-08: HLT. Association for Computational Linguistics, Columbus, Ohio, pp. 398–406.

Johnson, M., Goldwater, S., 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. Association for Computational Linguistics, Boulder, Colorado, p. 675.

Kamper, H., Settle, S., Shakhnarovich, G., Livescu, K., 2017. Visually grounded learning of keyword prediction from untranscribed speech. ArXiv170308136 Cs.

Le, V.B., 2006. Reconnaissance automatique de la parole pour des langues peu dotées (Theses). Université Joseph-Fourier - Grenoble I, Grenoble.

Lee, C., O'Donnell, T.J., Glass, J.R., 2016. Unsupervised Lexicon Discovery from Acoustic Input. TACL 3, 389–403.

Lehmann, C., 1999. Documentation of endangered languages A priority task for linguistics. ASSidUE Arbeitspapiere Semin. Für Sprachwiss. Univ. Erf.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, L., Dollár, P., 2014. Microsoft COCO: Common Objects in Context. ArXiv14050312 Cs.

Liu, C., Trmal, J., Wiesner, M., Harman, C., Khudanpur, S., 2017. Topic Identification for Speech without ASR. ArXiv170307476 Cs.

Ludusan, B., Synnaeve, G., Dupoux, E., 2015. Prosodic boundary information helps unsupervised word segmentation. Association for Computational Linguistics, pp. 953–963. doi:10.3115/v1/N15-1096

Ludusan, B., Versteegh, M., Jansen, A., Gravier, G., Cao, X.-N., Johnson, M., Dupoux, E., 2014. Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems. Presented at the Language Resources and Evaluation Conference.

Lyzinski, V., Sell, G., Jansen, A., 2015. An evaluation of graph clustering methods for unsupervised term discovery, in: INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015. ISCA, pp. 3209–3213.

Park, A., Glass, J.R., 2005. Towards unsupervised pattern discovery in speech. IEEE, pp. 53–58. doi:10.1109/ASRU.2005.1566529

Pellegrini, T., 2008. Transcription automatique de langues peu dotées. Université de Paris-Sud. Faculté des Sciences d'Orsay (Essonne), Paris.

Piantadosi, S.T., 2014. Zipf's word frequency law in natural language: A critical review and future directions. Psychon. Bull. Rev. 21, 1112–1130. doi:10.3758/s13423-014-0585-6

Polguère, A., 2001. Notions de base en lexicologie. Observatoire de Linguistique Sens-Texte, Montréal.

Räsänen, O., Doyle, G., Frank, M., 2015. Unsupervised word discovery from speech using automatic segmentation into syllable-like units. Presented at the Interspeech.

Roy, D.K., Pentland, A.P., 2002. Learning words from sights and sounds: a computational model. Cogn. Sci. 26, 113–146. doi:10.1207/s15516709cog2601_4

Sands, B., Miller, A.L., Brugman, J., 2007. The Lexicon in Language Attrition: The Case of N|uu. Presented at the 37th Annual Conference on African Linguistics, Cascadilla Proceedings Project, pp. 55–65.

Schwarz, D., 2007. Corpus-Based Concatenative Synthesis. IEEE Signal Process. Mag. 24, 92–104.

Signoret, C., 2010. Exploration des mécanismes non conscients de la perception de la parole : approches comportementales et électroencéphalographiques. Université Lumière Lyon 2, Lyon.

Tsunoda, T., 2006. Language Endangerment and Language Revitalization: An Introduction. Walter de Gruyter, Berlin.

UNESCO, n.d. FAQ on endangered languages | United Nations Educational, Scientific and Cultural Organization [WWW Document]. URL http://www.unesco.org/new/en/culture/themes/endangered-languages/faq-on-endangered-languages/ (accessed 11.22.16).

Versteegh, M., Anguera, X., Jansen, A., Dupoux, E., 2016. The Zero Resource Speech Challenge 2015: Proposed Approaches and Results. Procedia Comput. Sci. 81, 67–72. doi:10.1016/j.procs.2016.04.031

Liste des figures

Liste des tableaux

Table des annexes

1 Liste des images en Mboshi et occurrences des mots clefs

Les fichiers audio du corpus de Guy-Noël Kouarata n'étaient transcrits d'aucune manière que ce soit. De fait, nous n'avons aucun alignement entre le signal et le mot vedette de l'image. Nous avons donc écouté l'ensemble des fichiers audio et relevé pour chacun deux les timecodes de début et de fin des différentes occurrences du mot vedette.

Nous voulons toutefois souligner que les timecodes indiqués sont *approximatifs*. En effet, n'étant pas locuteur du Mboshi, la tâche de segmentation s'est avérée ardue puisque nous ne connaissons pas la morphologie de cette langue. Ainsi, il est possible que les timecodes ne correspondent pas à la vraie segmentation. De plus, nous avons inclus dans les timecodes, des mots dont la prononciation nous semblaient proche du mot vedette, suffisamment proche pour que ce soit le même (*ondongo* et *onlongo* ou bien encore *dongodongo* et *dongo* par exemple)

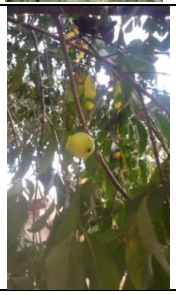
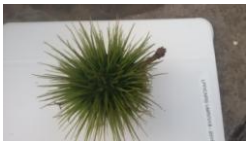
Le signe *H* correspond à des occurrences prononcées par le locuteur homme, et les signes *F1* et *F2* correspondent aux occurrences prononcées par la première locutrice ou par la seconde locutrice respectivement.

Image	Mot	Nom WAV	Timecodes *Loc Debut→Fin*	Notes
	Alerere	Elicit_0	H 01.958 → 02.267 H 08.354 → 08.663 F1 11.713 → 12.176 H 29.139 → 29.467 H 48.530 → 49.206 H 51.153 → 51.655 F1 56.756 → 57.220 H 62.303 → 62.767 H 65.221 → 65.704 H 88.779 → 89.455 H 105.948 → 106.412	
	Ambamba	Elicit_1	F1 01.293 → 02.148 H 02.683 → 03.271 H 04.501 → 04.329 F1 07.095 → 07.496 H 10.143 → 10.732 F1 17.382 → 17.810 H 24.067 → 24.549 H 31.761 → 32.242 F1 39.061 → 39.622 H 39.916 → 40.344 F1 40.772 → 41.280 H 44.805 → 45.286	

	Asuu	Elicit_2	H 02.083 → 02.413 F1 04.064 → 04.347 F1 04.418 → 04.866 H 06.140 → 06.588 F1 07.720 → 08.215 F1 08.805 → 09.159 H 11.375 → 11.729 H 17.031 → 17.362 H 47.090 → 47.444 F1 48.175 → 48.718 F1 48.718 → 49.236 F1 58.265 → 58.973 H 60.906 → 61.355 H 63.241 → 63.571 H 68.402 → 68.803 H 73.876 → 74.229 H 75.409 → 76.045	
	Aswebhe	Elicit_3	H 01.601 → 02.124 F1 03.012 → 03.558 H 04.286 → 04.741 H 06.835 → 07.199 H 08.564 → 08.906 H 23.077 → 23.532	
	Atembele	Elicit_4	*ND*	Nous n'avons pas entendu le mot *atembele* dans les signaux de parole
	Atsinga	Elicit_5	H 08.074 → 08.429 H 10.201 → 10.578 H 12.236 → 12.879 H 16.292 → 16.890 H 21.943 → 22.320 H 23.051 → 23.272 H 28.459 → 28.836 H 49.792 → 50.191 H 57.172 → 57.704 F1 59.830 → 60.229 H 60.606 → 61.049 H 63.509 → 63.974 H 72.170 → 72.479 H 72.699 → 73.008 H 73.162 → 73.420 H 74.743 → 74.978 H 75.007 → 75.507	

	Ateyri	Elicit_6	H 01.969 → 02.350 H 22.420 → 22.848 H 39.744 → 39.136 H 55.748 → 56.157 H 56.326 → 56.680	
	Ayaa	Elicit_7	H 03.172 → 03.436 F1 07.788 → 08.052 F1 40.354 → 40.728 H 43.541 → 43.849	
	Bange	Elicit_8	H 18.939 → 19.496 F1 20.053 → 20.471 H 20.889 → 21.354 H 29.249 → 29.806 H 31.060 → 31.478 H 32.407 → 32.779 H 33.150 → 33.661 H 42.795 → 43.084 F1 50.198 → 50.510	
	Bongo	Elicit_9	F1 03.292 → 03.923 H 04.908 → 05.400 H 05.662 → 06.016 H 06.031 → 06.385 F1 21.010 → 21.456 H 21.641 → 22.133 H 22.703 → 22.949 F1 23.549 → 23.949	
	Bongo	Elicit_10	F1 14.791 → 15.518 H 36.766 → 37.260 H 57.289 → 57.667 H 57.696 → 58.191 H 82.696 → 83.039	
	Bve b'indoo	Elicit_11	H 32.218 → 32.667 H 34.412 → 34.911	Les timecodes correspondent à la prononciation du terme *indoo*

	Bvele b'ifundu bvua	Elicit_12	H 12.876 → 13.349 H 32.918 → 33.391 F1 33.755 → 34.301 F1 40.598 → 41.143	L'ensemble des timecodes correspondent à la prononciation du mot *bvua*
	Bvue ba ikyee	Elicit_13	H 02.150 → 02.981 F1 03.547 → 04.279 H 22.101 → 22.833 H 23.066 → 23.831	
	Bvue ba tswa	Elicit_14	*ND*	Nous n'avons pas entendu le mot *bvue ba tswa* dans les signaux de parole
	dongodongo	Elicit_15	F1 00.852 → 01.422 H 02.601 → 03.092 H 07.475 → 07.927 H 11.619 → 12.052 H 19.501 → 20.090 H 70.236 → 70.924	Les timecodes correspondents à la prononciation du terme *dongo* ou *dongodongo*
	Dumeledzooni	Elicit_16	H 02.448 → 03.034 F1 03.403 → 04.077 H 04.154 → 04.892 H 35.029 → 35.831	
	Dzoa	Elicit_17	F1 04.769 → 05.256 F1 07.976 → 08.375 H 08.845 → 09.208 H 10.525 → 10.960 H 23.600 → 24.071 H 25.484 → 25.846 H 31.907 → 32.306 H 63.226 → 63.642 H 66.414 → 66.830	
	Dzonyo	Elicit_18	H 02.026 → 02.387 F1 02.529 → 02.978 H 05.077 → 05.384	

			H 35.033 → 35.448	
	Dzwaama	Elicit_19	F1 16.719 → 17.301 H 18.147 → 18.808 H 19.654 → 20.209	
	Ebvulu etoo	Elicit_20	H 01.457 → 01.924 H 03.242 → 03.642 H 04.560 → 05.010 H 20.532 → 20.949 H 23.067 → 23.351 H 23.434 → 23.751 H 37.340 → 37.657 H 38.908 → 39.258	L'ensemble des timecode correspondent à la prononciation du terme *etoo*
	Ebvulu ya lengale	Elicit_21	F1 40.592 → 41.230 H 42.233 → 42.916 F2 43.873 → 44.488 H 58.846 → 59.188 H 59.598 → 60.008 H 98.368 → 98.892 H 101.489 → 102.309	L'ensemble des timecodes correspondent à la prononciation du terme *lengale*
	Ebvuma y'obvwese	Elicit_22	H 04.786 → 05.140 H 06.224 → 06.645 F2 07.369 → 08.187 F2 20.258 → 20.723 H 22.991 → 23.289 H 23.947 → 24.394 H 27.313 → 27.666 H 27.963 → 28.317 F2 37.736 → 38.089 H 57.461 → 57.963 H 75.977 → 76.498 H 78.676 → 79.290 H 83.808 → 84.366 F2 90.858 → 91.453 H 99.474 → 99.790	L'ensemble des timecodes correspondent à la prononciation du terme *ebvuma*
	Edzesa	Elicit_13	F2 16.236 → 16.777 H 16.660 → 17.036 H 17.812 → 18.306 H 18.753 → 19.247 H 19.388 → 19.788 H 20.212 → 20.659 H 24.964 → 25.434	

			H	70.306 → 70.706	
			H	71.295 → 71.624	
			H	74.465 → 74.841	
			H	75.288 → 75.782	
	Ekongo	Elicit_24	H	09.948 → 10.372	
			H	15.356 → 15.780	
			H	16.163 → 16.586	
			H	24.616 → 25.301	
			H	25.281 → 25.906	
			H	26.128 → 26.551	
			H	46.511 → 47.015	
			H	52.317 → 52.721	
			H	60.669 → 61.133	
			H	61.859 → 62.242	
			H	63.533 → 63.936	
			H	70.327 → 70.932	
	Elolo	Elicit_25	F2	01.613 → 02.255	
			H	02.977 → 03.538	
			H	10.945 → 11.399	
			H	36.601 → 36.975	
			H	36.922 → 37.269	
			H	41.975 → 42.456	
			H	45.818 → 46.352	
			F2	47.850 → 48.384	
			H	49.026 → 49.534	
			H	50.015 → 50.470	
			H	50.871 → 51.326	
			H	64.419 → 64.820	
			H	72.173 → 72.441	
			H	74.973 → 75.374	
			F1	75.855 → 76.176	
			H	79.545 → 79.946	
			F1	81.203 → 81.738	
			H	81.845 → 82.246	
			F2	86.283 → 86.765	
			H	89.978 → 87.433	
			F2	87.246 → 87.834	
	Esalaa y'ibia	Elicit_26	H	03.499 → 03.896	Les timecodes en gras correspondent à la prononciation du mot *esalaa.* Celui en gras souligné correspond à l'ensemble du terme *esalaa y'ibia*. Les autres correspondent à la
			F1	04.947 → 05.305	
			H	05.622 → 05.940	
			H	05.801 → 06.257	
			H	12.352 → 12.690	
			F2	15.527 → 15.944	
			H	16.500 → 16.837	
			F1	31.408 → 31.766	
			F2	43.447 → 44.142	
			F1	53.070 → 53.626	
			F2	53.664 → 54.102	

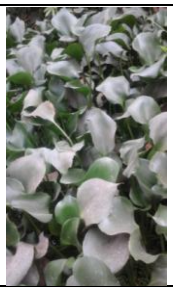
			F1 54.479 → 54.995 **H 55.154 → 55.689** **H 56.364 → 56.880** **F1 57.059 → 57.654** F2 62.594 → 63.030 **F2 64.230 → 64.726** **H 65.004 → 65.361**	prononciation de *(y')ibia* seulement
	Esie	Elicit_27	F2 01.379 → 01.911 H 02.241 → 02.774 F1 22.822 → 23.228 H 24.522 → 24.928 H 26.268 → 26.649 H 62.185 → 62.591 H 74.495 → 74.951 H 85.758 → 86.190 H 94.882 → 95.338	
	Esondo ya ondongo	Elicit_28	**H 02.661 → 03.181** **F1 03.875 → 04.644** F1 04.148 → 04.473 H 05.189 →0 5.610 H 07.048 → 07.544 F1 09.626 → 10.122 H 13.354 → 13.875 H 16.155 → 16.626 H 24.138 → 24.609 F1 56.112 → 50.608 H 58.380 → 58.876 H 60.463 → 60.884 H 119.135 → 119.705	Les timecodes en gras correspondent à l'occurence du mot *esondo*. Les autres correspondent aux occurrences du mot *ondongo/onlongo*
	Kongo a sika	Elicit_30	F2 01.792 → 02.645 H 03.950 → 04.994 H 06.184 → 07.128 H 09.667 → 10.420 H 11.877 → 12.690	

Tableau 18 Liste des images et occurrences des mots-clefs

2 Comparaison des images des différents corpus

Vision stéréotypique d'un objet	   
Vision stéréotypique d'une scène	  
Vison non stéréotypique d'un objet ou d'une scène	   

Tableau 19 Images issues de MSCOCO

Vision stéréotypique d'un objet	  
Vision stéréotypique d'une scène	   
Vison non stéréotypique d'un objet ou d'une scène	 

Tableau 20 Images issues du corpus Mboshi

Vision stéréotypique d'un objet	  
Vision stéréotypique d'une scène	 
Vison non stéréotypique d'un objet ou d'une scène	 

Tableau 21 Images issues du corpus Tima

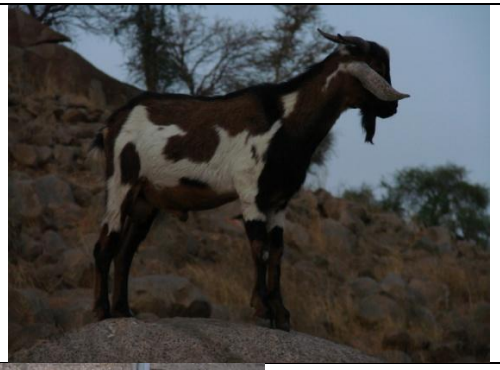
Vision stéréotypique d'un objet	 
Vision stéréotypique d'une scène	
Vison non stéréotypique d'un objet ou d'une scène	

Tableau 22 Images issues du corpus Tabaq

3 Disfluences possibles en fonction des locuteurs

Les cases vertes correspondent à une disfluence prononçable par le locuteur et les cases rouge à une disfluence imprononçable par le locuteur.

« --- » désignent les disfluences que nous avons exclues car elles n'étaient pas suffisamment réalistes compte tenu de leur position.

		er	uh	um	huh	er oh	uh oh	um oh	huh oh	er ah	uh ah	um ah	huh ah	oh	ah
Paul	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Bronwen	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Judith	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Elizabeth	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Bruce	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Jenny	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Amanda	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---
Phil	Début														
	Milieu														
	Fin					---	---	---	---	---	---	---	---	---	---

Tableau 23 Disfluences possibles en fonction du locuteur

4 Exemple d’un fichier de métadonnées au format JSON


```
{
   "disfluency":
   [
      "None",
      "None"
   ],
   "synthesisedCaption":"This wire metal rack holds several pairs of shoes and sandals.",
   "timecode":
   [
        [0.0, "WORD", "This"], [0.0, "SYL", ""], [0.0, "PHO", "#"], [9.6023, "PHO", "dh"], [77.7273, "PHO", "i"],
 [138.4091, "PHO", "s"],       [215.9659, "SYL", ""], [215.9659, "SEPR", " "], [215.9659, "WORD", "wire"], [215.9659,
 "PHO", "w"], [273.6364, "PHO", "ai"], [419.2045, "SYL", ""], [419.2045, "PHO", "@@"], [473.6932, "SYL", ""], [473.6932,
  "SEPR", " "], [473.6932, "WORD", "metal"], [473.6932, "PHO", "m"], [548.0682, "PHO", "e"], [641.3068, "SYL", ""],
 [641.3068, "PHO", "t"], [751.875, "PHO", "l="], [896.3636, "SYL", ""], [896.3636, "SEPR", " "], [896.3636, "WORD",
 "rack"], [896.3636, "PHO", "r"], [945.4545, "PHO", "a"], [1028.1818, "PHO", "k"], [1098.9205, "SYL", ""], [1098.9205,
 "SEPR", " "], [1098.9205, "WORD", "holds"], [1098.9205, "PHO", "h"], [1179.3182, "PHO", "ou"], [1266.7045, "PHO",
 "l"], [1328.8636, "PHO", "d"], [1356.9318, "PHO", "z"], [1420.4545, "SYL", ""], [1420.4545, "SEPR", " "], [1420.4545,
 "WORD", "several"], [1420.4545, "PHO", "s"], [1509.2614, "PHO", "e"], [1602.3864, "PHO", "v"], [1686.9318, "SYL", ""],
 [1686.9318, "PHO", "r"], [1718.1818, "PHO", "@"], [1762.1591, "PHO", "l"], [1818.0682, "SYL", ""], [1818.0682, "SEPR",
 " "], [1818.0682, "WORD", "pairs"], [1818.0682, "PHO", "p"], [1901.4205, "PHO", "e@"], [2116.3636, "PHO", "z"],
 [2168.8068, "SYL", ""], [2168.8068, "SEPR", " "], [2168.8068, "WORD", "of"], [2168.8068, "PHO", "@"], [2244.375,
 "PHO", "v"], [2292.5, "SYL", ""], [2292.5, "SEPR", " "], [2292.5, "WORD", "shoes"], [2292.5, "PHO", "sh"], [2417.0455,
 "PHO", "uu"], [2679.1477, "PHO", "z"], [2761.7614, "SYL", ""], [2761.7614, "SEPR", " "], [2761.7614, "WORD", "and"],
 [2761.7614, "PHO", "@"], [2814.0909, "PHO", "n"], [2875.6818, "PHO", "d"], [2911.9318, "SYL", ""], [2911.9318, "SEPR",
 " "], [2911.9318, "WORD", "sandals"], [2911.9318, "PHO", "s"], [2997.8409, "PHO", "a"], [3080.6818, "PHO", "n"],
 [3169.6591, "SYL", ""], [3169.6591, "PHO", "d"], [3207.2727, "PHO", "l="], [3387.2159, "PHO", "z"], [3634.8864, "PHO",
 "#"], [3648.6364, "SYL", ""], [3648.6364, "PHO", "#"], [3927.75, "SIL", ""], [3927.75, "PUNCT", "."], [3927.75, "PHR",
 ""]
   ],
   "speed":1.1,
   "imgID":42,
   "speaker":"Bronwen",
   "captionID":641613,
   "wavFilename":"42_641613_Bronwen_None_1-1.wav",
   "duration":3.92775
}
```


Figure 26 Exemple du stockage de métadonnées au format JSON

5 Segments étudiés pour les calculs de DTW

Nous avons indiqué en gras les mots auxquels nous nous sommes intéressés pour faire les calculs de DTW. Nous avons corrigé la légende 145464 pour laquelle il manquait un « t » au mot « kitchen » afin que la prononciation soit correcte. Nous avons exclu la légende 725123 car celle-ci contenait le mot « flower » et non « flowers ».

ID Image	ID Légende	Légende
109229	89335	A young boy throwing a **frisbee** in a grassy field
	89524	a young boy in the park throwing a **frisbee**
	90802	A young boy throws a **frisbee** in a tree lined park.
	91756	A kid in a city park throws a bright green **Frisbee**.
	92593	A boy throwing a green **frisbee** in a grass field.
393266	102294	A car driving through a **tunnel** under buildings
	108060	Car passing through a very small **tunnel** in a city street.
	109143	A car driving through a **tunnel** between two buildings.
	116697	A **tunnel** in the middle of a street with a car about to go down it
	144381	A car is driving through a **tunnel** in a city street.
262242	549896	A woman standing on a **tennis** court holding a racquet.
	556412	A woman picking up a ball with a **tennis** racquet.
	556508	Three people with racquets stand on a **tennis** court.
	556994	a person with a **tennis** racket picks up a **tennis** ball
	560729	A woman tries to scoop a **tennis** ball with her racket.
524392	131640	A **kitchen** scene with focus on the refrigerator.
	136326	A fridge that is silver and in a **kitchen**.
	136791	A **kitchen** contains a large refrigerator with a freezer below it.
	137106	Stainless steel fridge in the **kitchen** of a home.
	137814	A **kitchen** with a large, newer looking refridgerator (*sic*).
262262	660653	A tall **tower** with a clock stands above a winter sky.
	678035	There is a tree next to the clock **tower**.
	678068	A large clock **tower** on a cloudy winter day.
	679334	There is a clock in the center of a **tower**.
	681881	a **tower** that will have a large clock at the top
262275	44907	A young **girl** in a bright dress rides a horse.
	47583	A **girl** in a pink dress sitting on top of a horse.
	48297	A little **girl** riding on a brown horse.
	49254	A little **girl** holds onto her saddle as she sits on a horse.
	50169	A young **girl** wearing a helmet riding a horse.
133	420587	A loft **bed** with a dresser underneath it.
	421694	A **bed** and desk in a small room.
	421754	Wooden **bed** on top of a white dresser.

	423389	A **bed** sits on top of a dresser and a desk.
	424250	Bunk **bed** with a narrow shelf sitting underneath it.
143	483949	**Birds** perch on a bunch of twigs in the winter.
	487240	A number of small **birds** sitting at the top of a bare tree.
	491767	Many **birds** perched on the limbs of a tree.
	492805	Eight **birds** perch on a branch on an overcast day.
	496462	Several **birds** are sitting on small tree branches.
164	117663	A small **kitchen** with low a ceiling
	145464	A small **kitchen** area with a sunlight and and angled ceiling. (*sic*)
	173583	an image of a **kitchen** loft style setting
	205026	a small **kitchen** with a lot of filled up shelves
	213867	A **kitchen** with a slanted ceiling and skylight.
131295	434531	An aerial view of a **train** on **train** tracks.
	434606	A **train** on a **train** track next to a grassy area.
	435188	a top view of a **train** riding on some tracks
	440900	The aerial photo shows a **train** viewed from above.
	442172	A **train** runs parallel to the other tracks.
192047	564562	A white **sink** in a corner underneath a small mirror and light.
	564661	A white **sink** sitting under a mirror in a bathroom.
	566884	An all white bathroom consisting of a mirror and **sink** view.
	567961	A bathroom with **sink**, mirror, and lights in it.
	570700	An all-white bathroom with a light fixture and small mirror above a **sink**
262394	100797	**Motorcycle** parked along the street next to a building.
	107781	a **motorcycle** parked next to other motorcycles on city street
	107967	A **motorcycle** is parker on a city sidewalk.
	108234	a black **motorcycle** is on display with more on either side
	112017	A parked **motorcycle** is parked on a city street.
262396	522552	A cat is sitting on top of a computer **chair** which is covered in hair.
	523491	A long haired cat on the top of an office **chair** covered in cat hair.
	523959	A cat laying on the top of a **chair** that is covered in cat hair.
	534459	A cat is sitting atop of a hair **chair**.
	537423	A view of a cat getting fur all over a **chair**.
257	488960	There is a small bus with several **people** standing next to it.
	491417	**people** standing besides a bus taking to each other
	495413	**People** eating from food trucks near a commemorative archway
	498524	Several **people** walking on a sidewalk near a large arch with figures on it.
	499340	A very big pretty arch way with a bunch of **people** near it.
393478	488839	The green and yellow **train** is rounding the bend of a track.
	489139	A green and yellow **train** going up an incline in the snow.
	490087	A **train** is traveling on the railroad next to the snow.
	491341	A **train** coming out of an enclosure under a snowy mountain.
	497512	there is a **train** coming up the tracks
524333	747766	A group of **people** sitting around a table.

	750678	this is a group of **people** eating a meal
	754276	a group of **people** sitting around a big restaurant table
	755968	A group of **people** sitting at a table that has food on it.
	759707	The group of **people** are sitting together eating.
294	549895	A man standing in front of a microwave next to **pots** and pans.
	556411	A man displaying **pots** and utensils on a wall.
	556507	A man stands in a kitchen and motions towards **pots** and pans.
	556993	a man poses in front of some **pots** and pans
	560728	A man pointing to **pots** hanging from a pegboard on a gray wall.
262466	717923	A bunch of **flowers** sticking out of a glass vase.
	724538	THIS IS A VERY PRETTY PICTURE OF A VASE OF **FLOWERS**
	725123	a flower that is in some kind of jar
	726278	A vase full of **flowers** sitting beside a blue piece of pottery.
	730175	A fall arrangement of **flowers** has a raffia bow.
87435	607351	The red **bus** has an anime girl on it.
	609127	Anime themed **bus** traveling down a tree lined street.
	611923	a red blue and white **bus** riding down a empty street
	622237	The back of a **bus** in Japan with a Sailor Moon character on it.
	626689	A photo of the back of a public transportation **bus** with anime advertisement.
21900	521131	A man riding a **horse** in front of a fence.
	521230	a man on a white **horse** in a stable ridding
	524542	a man on a **horse** standing in side a white fence
	527425	A person in a fancy outfit riding on a big pretty **horse**.
	531043	A man riding on the back of a white **horse**.

Tableau 24 Mot étudiés pour les calculs de variabilité inter- et intra-locuteurs

6 Variabilité intra-locuteur en detail

Comme nous l'avons déjà mentionné, lorsque nous comparons les segments d'un même locuteur entre eux, nous obtenons les valeurs de DTW les plus faibles puisque la variabilité intralocuteurs est plus faible que la variabilité interlocuteurs. Nous avons donc voulu comparer plus précisément la variabilité intralocuteurs des différentes voix en fonction des mots.

Nous pouvons constater grâce au tableau de la page suivante que les voix de synthèse et notre propre voix varient d'une manière sensiblement identique et ont une variabilité forte ou faible pour les mêmes mots. Notre voix ne présente pas de variabilité plus forte que celles des autres locuteurs.

Par exemple, l'ensemble des voix présente une variabilité faible pour *motorcycle* (ID image 262394) et *tunnel* (ID image 393266). Cela s'explique par le fait que les occurrences de ces deux mots ont un contexte gauche et droit quasiment identique et se situent dans des groupes prosodiques identiques.[30]

Motorcycle parked along the street next to a building.	ˈ**moʊtərˌsaɪkəl** **p**ɑrkt əˈlɔŋ ðə strit nɛkst tu ə ˈbɪldɪŋ.
a **motorcycle** parked next to other motorcycles on city street	**ə** ˈ**moʊtərˌsaɪkəl** **p**ɑrkt nɛkst tu ˈʌðər ˈmoʊtərˌsaɪkəlz ɑn ˈ sɪtistrit
A **motorcycle** is parker on a city sidewalk.	**ə** ˈ**moʊtərˌsaɪkəl** **ɪ**z ˈpɑrkər ɑn ə ˈsɪti ˈsaɪˌdwɔk.
a black **motorcycle** is on display with more on either side	ə blæ**k** ˈ**moʊtərˌsaɪkəl** **ɪ**z ɑn dɪˈspleɪ wɪð mɔr ɑn ˈiðər saɪd
A parked **motorcycle** is parked on a city street.	ə pɑrk**t** ˈ**moʊtərˌsaɪkəl** **ɪ**z pɑrkt ɑn ə ˈsɪti strit.

Tableau 25 Contexte d'occurrence du mot *motorcycle*

A l'inverse, on peut constater une forte variabilité pour le mot *chair* (ID image 262396) puisque les occurrences des mots apparaissent dans des contextes relativement différents et dans des blocs prosodiques différents. [30]

A cat is sitting on top of a computer **chair** which is covered in hair.	ə kæt ɪz ˈsɪtɪŋ ɑn tɑp ʌv ə kəmˈpjutə**r** **ʧɛr** **w**ɪʃ ɪz ˈk ʌvərd ɪn hɛr.
A long haired cat on the top of an office **chair** covered in cat hair.	ə lɔŋ hɛrd kæt ɑn ðə tɑp ʌv ən ˈɔfə**s** **ʧɛr** ˈ**k**ʌvərd ɪn kæt hɛr.
A cat laying on the top of a **chair** that is covered in cat hair.	ə kæt ˈleɪɪŋ ɑn ðə tɑp ʌv **ə** **ʧɛr** **ð**æt ɪz ˈkʌvərd ɪn k æt hɛr.
A cat is sitting atop of a hair **chair**.	ə kæt ɪz ˈsɪtɪŋ əˈtɑp ʌv ə hɛ**r** **ʧɛr**.
A view of a cat getting fur all over a **chair**.	ə vju ʌv ə kæt ˈgɛtɪŋ fɜr ɔl ˈoʊvər **ə** **ʧɛr**.

Tableau 26 Contexte d'occurrence du mot *chair*

Pour conclure, la variabilité interlocuteurs et intralocuteurs sont dans les normes de ce que l'on pourrait obtenir avec de vraies voix. De ce fait, nous pouvons considérer que notre corpus est suffisamment réaliste pour des études en apprentissage automatique ou en découverte non supervisée de mots.

	Bronwen	gTTS	Phil	Amanda	Bruce	Paul	Jenny	William	Elizabeth	Judith
133 - bed	40,04555	51,21985	48,25326	58,24821	38,70067	50,97665	42,49177	42,85616	45,97121	44,53298
143 - birds	36,75739	39,43098	47,28181	40,04096	38,78028	37,70544	41,36417	45,35146	33,03115	44,94935
164 - kitchen	33,90203	43,26965	33,11462	34,41357	31,17463	29,65721	28,80343	40,75512	29,77964	37,66948
257 - people	43,07599	46,77444	37,36075	41,46293	44,04889	40,15439	44,53071	63,27563	38,78862	51,33624
294 - pots	30,85749	34,95873	39,4952	32,5764	30,2702	35,14963	24,64781	41,84347	31,79009	28,72233
21900 - horse	47,81121	56,40103	64,7259	39,10707	53,1255	54,43812	42,11468	59,02238	50,81599	58,8196
87435 - bus	41,45508	48,81287	46,51145	40,17989	40,28452	51,2833	37,62604	49,5325	34,71214	39,80888
109229 - frisbee	43,50818	51,02595	54,17458	46,53002	46,66309	49,85501	47,1443	40,63173	42,34054	43,0073
131295 - train	39,12198	42,03735	44,84742	34,52897	34,10387	42,75778	37,27265	36,99578	23,92085	51,50624
192047 - sink	54,04981	60,30008	45,7033	47,594	43,86865	41,9619	44,3879	47,12777	43,49227	60,47364
262242 - tennis	31,74806	40,42976	36,47575	29,20661	35,85765	35,63	34,76055	33,36725	32,63238	45,51942
262262 - tower	59,1313	46,23891	65,6581	41,19938	48,04086	61,7707	49,07852	59,72313	44,8047	56,90283
262275 - girl	30,09599	48,59764	40,89224	30,3326	33,98326	27,24196	42,03391	52,4439	27,20252	45,42955
262394 - motorcycle	16,89626	36,8184	37,62013	17,51475	23,07385	14,85786	14,36666	43,15993	18,66057	35,48132
262396 - chair	50,12299	74,15871	62,20445	66,65047	62,05593	66,58071	66,80781	43,64789	47,99336	51,08923
262466 - flowers	31,06669	44,25142	48,27576	42,1336	31,90025	28,65056	34,82723	60,00436	31,2245	53,57811
393266 - tunnel	18,95193	35,45072	25,61508	16,20675	28,77609	19,69304	18,17098	41,03862	19,14173	60,14875
393478 - train	34,03904	39,08676	49,90756	33,20227	33,0566	36,64197	38,36408	40,37634	25,26209	47,7571
524333 - people	23,71422	26,89512	28,54276	29,17613	29,05205	38,24883	30,33954	41,48827	23,28107	39,15809
524392 - kitchen	41,4256	49,22831	54,74328	45,29979	40,22352	47,57306	46,93989	46,66499	36,03862	53,98998
Moyenne	37,38884	45,76933	45,57017	38,28022	38,35202	40,54141	38,30363	46,46533	34,0442	47,49402

Tableau 27 Valeur de DTW intra-locuteur

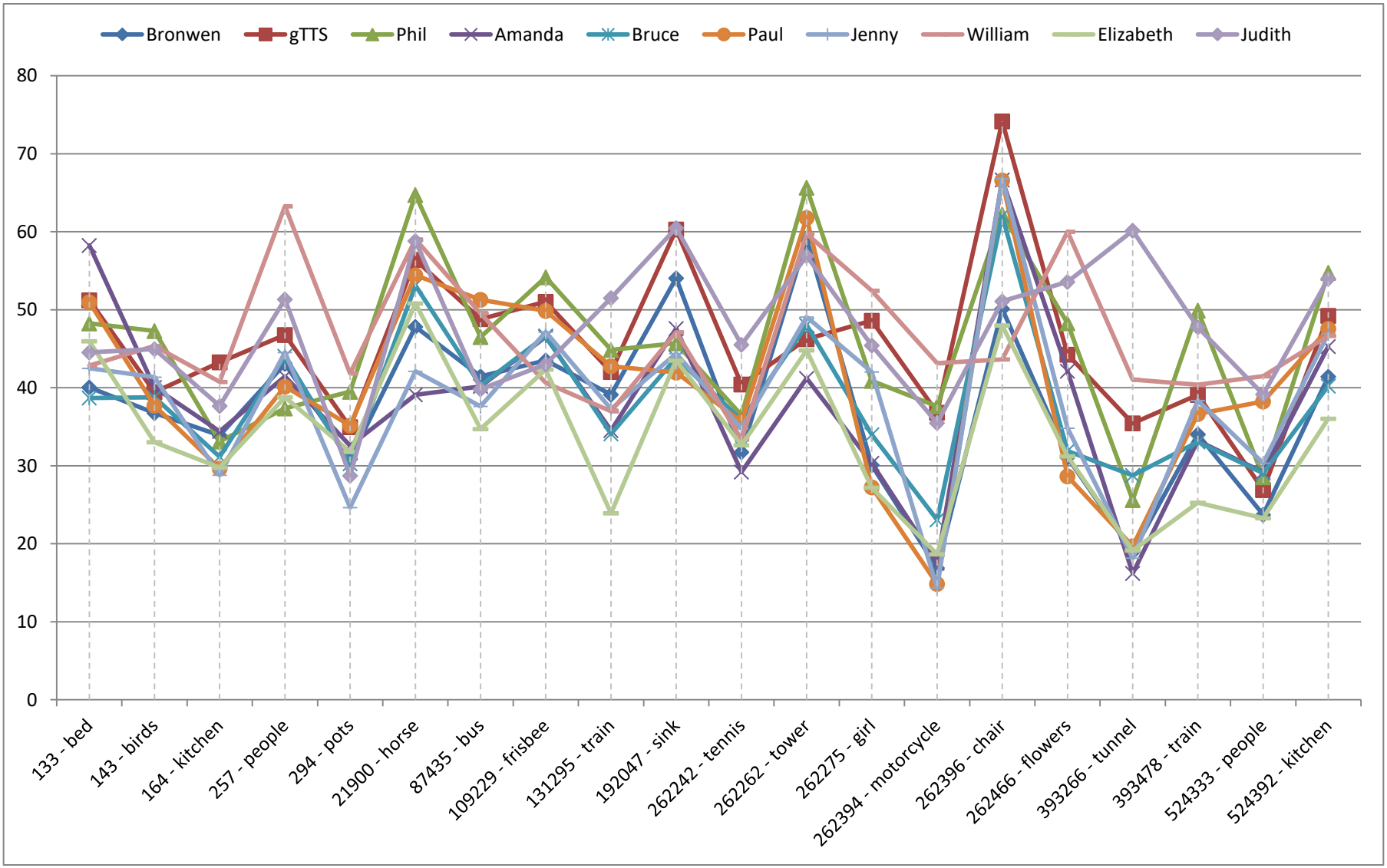


Tableau 28 Variabilité intra-locuteur pour chacun des mots étudié

7 Problèmes rencontrés lors de la synthèse du corpus

Nous avons rencontré quelques problèmes lors de la création de ce corpus.

Le premier problème que nous avons rencontré venait de SOX qui affichait parfois ce message d'erreur : « *Header SOX « sox WARN wav : Premature EOF on .wav input file* ». Ce problème n'est pas de notre fait mais d'un bogue lors la création du fichier WAV par le système de Voxygen. Celui-ci indique que la durée totale du fichier WAV, annoncée dans le *header* de celui-ci, est supérieure à la durée réelle. Malgré l'affichage de cette erreur, l'opération d'accélération ou de ralentissement était correctement réalisée. Cela pose toutefois un autre problème, la durée *duration* ainsi que les derniers *timecode* (fin de syllabe, fin de silence et fin de phrase) étaient incorrects. Ainsi, pour corriger cela, nous avons recalculé de manière systématique la durée de chacun des fichiers WAV produits par Voxygen. En effet, avoir la durée correcte de chacun des fichiers s'avère indispensable si l'on souhaite les concaténer afin de former un fichier plus long et que les *timecodes* des fichiers suivants soient mis à jour correctement.

Le second problème que nous avons rencontré provenait du layer *HTTP* du système de Voxygen qui générait parfois cette erreur « *HTTP Error 500: Internal Server Error* ». Cette erreur était imprévisible, mais très dérangeante puisqu'elle bloquait le *pipeline* de synthèse. Afin que cette erreur ne perturbe pas le processus de synthèse, dès qu'une exception de ce type était levée par Python, nous redémarrions le layer *HTTP*. Cette opération était suffisante pour permettre de poursuivre le processus de synthèse.

Le dernier problème que nous avons rencontré était de notre fait. En effet, afin d'ajouter les disfluences du milieu aux bons endroits (soit avant un nom soit après un déterminant) nous avons tokenisé l'ensemble des légendes par TreeTagger. Une fois la disfluence ajoutée, il était nécessaire de raccrocher les tokens entre eux afin de faire une phrase. TreeTagger ne gardant pas la trace des séparateurs, nous avons décidé de les coller par un espace. Cela posait problème puisque TreeTagger tokenisait « isn't » en « is » et « n't ». Réunir ces tokens par un espace était évidemment faux et générait des problèmes lors de la synthèse, où le terme « apostrophe » était synthétisé. Pour corriger ce problème nous avons donc écrit une simple fonction permettant de retrouver les séparateurs à partir des tokens et de la version non tokenisée.

8 Script de manipulation du corpus

Notre script Python permet de facilement manipuler le corpus. En effet, le corpus que nous avons créé est conséquent (plus de 600.000 légendes). Il est indispensable que les personnes qui souhaiteraient l'utiliser puissent trouver facilement ce qu'elles désirent.

Le script a été écrit en tirant parti du modèle objet de Python.

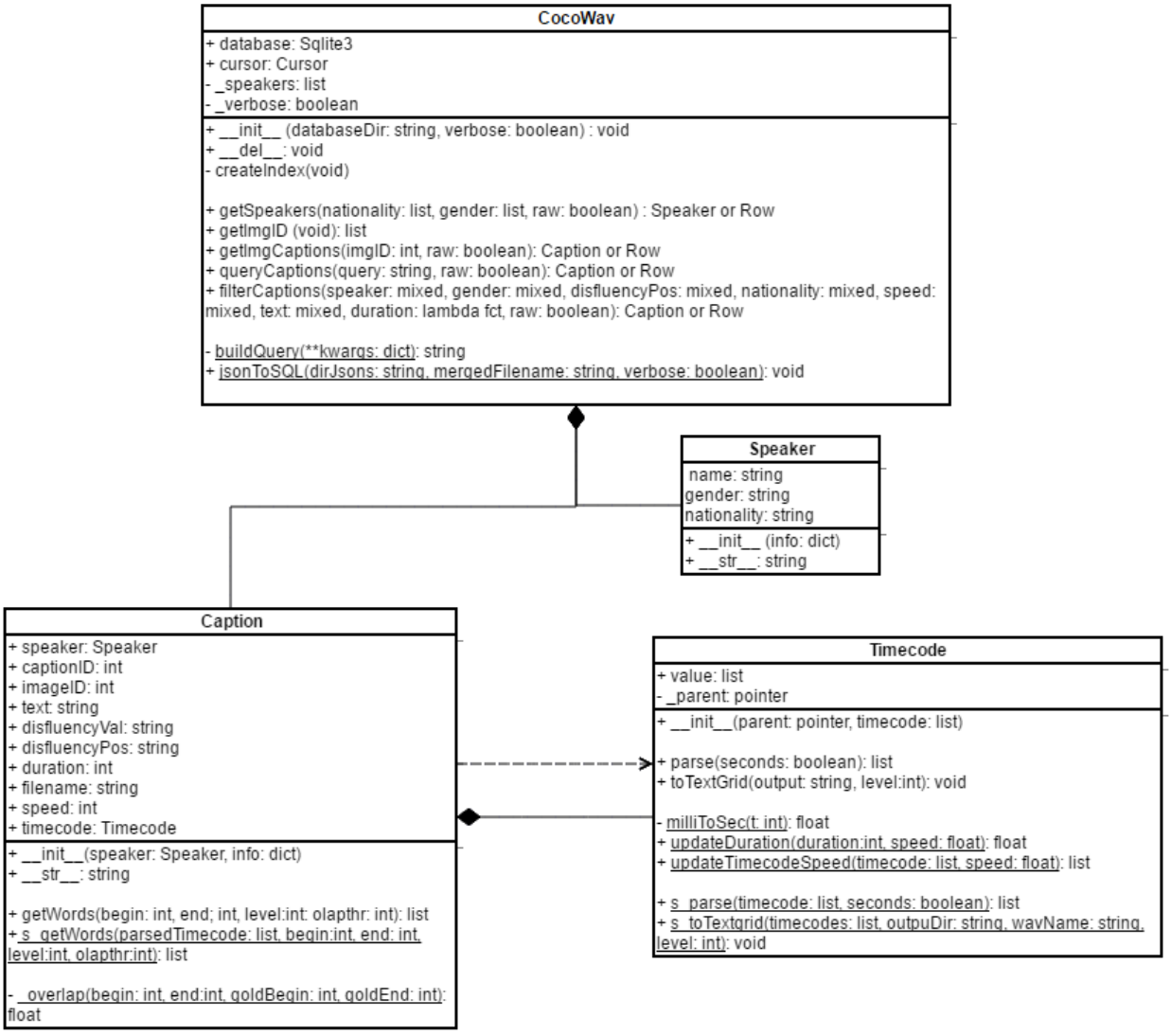


Figure 27 Diagramme UML du script cocoWav_API.py

8.1 Classe principale *CocoWav*

La classe principale est *CocoWav* et prend comme seul paramètre d'instanciation le chemin vers la base de données. Si elle existe, nous nous connectons à celle-ci et l'utilisateur peut alors commencer à faire des requêtes sur le corpus. Au cas où l'utilisateur n'aurait pas téléchargé la base de données déjà existante, mais seulement les fichiers JSON indépendants, cette même classe dispose d'une méthode statique (*jsonToSQL)* permettant de fusionner tous les fichiers JSON dans une base de données SQLite.

La méthode la plus importante de cette classe est la méthode *filterCaptions* qui permet de récupérer les légendes qui intéressent l'utilisateur en fonction des filtres de son choix : nom du locuteur, sexe du locuteur, position d'une disfluence, nationalité du locuteur, vitesse du fichier WAV, durée du fichier WAV ou bien encore sur la présence spécifique d'un mots ou groupe de mot dans la légende. La méthode fait ensuite appel à la méthode privée statique *_buildQuery* afin de construire dynamiquement la requête SQL correspondant au filtre de l'utilisateur. Par exemple, la requête suivante permet à l'utilisateur de récupérer toutes les légendes contenant le mot « keys » prononcées par un locuteur homme de nationalité américaine et dont la vitesse a été ralentie de 10% :

```
db.filterCaptions(gender="Male", nationality="US", speed=0.9, text='%keys%')
```
[49]

Notre script génère automatiquement la requête suivante :

```
SELECT * FROM captions INNER JOIN speakers ON
captions.speaker=speakers.name WHERE (gender="Male") AND
(nationality="US") AND (speed="0.9") AND (text LIKE "%keys%")
```

Si l'utilisateur a besoin de faire une requête vraiment spécifique et que la méthode *filterCaptions* ne permettait pas de la faire, nous avons créé une méthode *queryCaptions* où l'utilisateur peut donner comme argument sa propre requête SQL.

Ces deux méthodes renvoient à l'utilisateur des objets de type *Caption* que nous détaillerons par la suite. Etant donné que la création d'objets peut être très lourde si les résultats de la requête sont nombreux, nous avons ajouté à ces deux méthodes l'argument facultatif `raw=False`. Si l'argument est à « False » nous renvoyons des objets de type *Caption*. Dans le cas contraire, nous renvoyons directement les objets de type *Row* de SQLite. Si tel est le cas, l'utilisateur peut accéder aux différents champs retournés en

[49] *db* étant un objet du type CocoWav

spécifiant explicitement leur titre : *resultat['captionID']* pour avoir l'ID de la légende. Récupérer les résultats bruts permet de récupérer les résultats plus rapidement que si des objets de type *Caption* sont créés.

Cette classe dispose de trois autres méthodes *getSpeakers*, *getImgID* et *getImgCaptions* qui permettent respectivement de récupérer tous les locuteurs du corpus, tous les ID des images du corpus et toutes les légendes appartenant à une image en particulier.

8.2 Classe *Caption*

La classe *Caption* permet de créer des objets contenant toutes les informations sur une légende en particulier : ID de l'image, ID de la légende, texte de la légende, position de la disfluence, valeur de la disfluence, durée du fichier WAV, nom du fichier WAV, vitesse du fichier WAV, le locuteur (référence vers un objet *Speaker*) et les timecodes (objet de type *Timecode*). C'est un objet de ce type qui est renvoyé à l'utilisateur lorsqu'il fait une requête sur le corpus. Cela lui permet de facilement accéder à toutes les informations sur la légende en faisant par exemple *caption.filename*, *caption.text*, *caption.disfluencyPos*, etc.

Cette classe dispose d'une méthode *getWords* qui permet de retourner à l'utilisateur les mots, syllabes et phonèmes compris entre deux indications temporelles fournis par l'utilisateur. Par exemple, la ligne de code suivante permet à l'utilisateur de récupérer tous les mots, syllabes et phonème situé entre la seconde 1,5 et 3,5 :

```
caption.getWords(1.5, 3.5, seconds=True, level=3, olapthr=50)
```

L'argument *seconds* permet d'indiquer que les données temporelles qui sont indiquées sont en secondes. Par défaut, elles doivent être données en millisecondes. L'argument *level* permet d'indiquer exactement ce que l'on veut récupérer. Ainsi, si *level* vaut 1 on ne récupère que les mots, s'il vaut 2 on récupère les mots et les syllabes et s'il vaut trois on récupère les mots, les syllabes et les phonèmes.

Le dernier argument *olapthr* permet de retourner les mots dont au minimum 50% de leur durée totale est incluse dans l'intervalle indiqué par l'utilisateur. Par exemple, dans l'exemple précédent, s'il y avait un mot qui débutait à la seconde 0,5 et finissait à la seconde 1,5, il serait retourné dans les résultats car 50% de sa longueur totale rentre dans l'intervalle temporel donné par l'utilisateur. Par contre, si le mot était de 10 secondes plus court, il ne serait pas retourné, car seulement 40% de sa durée totale se

trouverait dans l'intervalle donné par l'utilisateur. La durée de recouvrement du segment par rapport à l'intervalle de l'utilisateur est calculée par la méthode privé statique *_overlap*[50].

La méthode statique *s_getWords* permet de faire la même chose que *getWords* mais ne nécessite pas que l'utilisateur est instancié un objet de type *Caption.*

L'ensemble de ces deux méthodes se base sur retour de la méthode *parse* de la classe *Timecode* que nous présenterons ensuite.

8.3 Classe *Timecode*

La classe *Timecode* permet de facilement gérer les *timecodes* et savoir à quel moment un mot, une syllabe ou un phonème a été prononcé.

La méthode *parse* permet de parser les *timecodes* et d'indiquer précisément le début et la fin de chaque mot, syllabe et phonème. En effet, ceux-ci ne le spécifient pas explicitement. Ainsi nous transformons des *timecode* en une version plus structurée, où il est explicitement spécifié quels phonèmes composent les syllabes, et quelles syllabes composent les mots. Nous avons également créé une version statique de la méthode afin que les *timecodes* obtenus sans initialiser d'objet puissent également être parsés. Celle-ci se nomme *s_parse*. Par défaut, notre méthode renvoie les durées en millisecondes, toutefois il est également possible de les récupérer en secondes si l'utilisateur le souhaite.

- Timecode d'entrée :

```
...
[1926.3068, "SYL", ""],
[1926.3068, "SEPR", " "],
[1926.3068, "WORD", "white"],
[1926.3068, "PHO", "w"],
[2050.7955, "PHO", "ai"],
[2144.6591, "PHO", "t"],
[2179.3182, "SYL", ""],
[2179.3182, "SEPR", " "]
...
```

[50] Le code permettant de faire le calcul a été trouvé sur http://stackoverflow.com/ [consulté le 24 mai 2017]. Le lien précis de la source est indiqué en commentaire sous la définition de la méthode.

- Timecodes de sortie

```
...
 {
 'begin': 1926.3068,
 'end': 2179.3182,
 'syllable': [
              {'begin': 1926.3068,
              'end': 2179.3182,
              'phoneme': [
                          {'begin': 1926.3068,
                           'end': 2050.7955,
                           'value': 'w'},
                          {'begin': 2050.7955,
                           'end': 2144.6591,
                           'value': 'ai'},
                          {'begin': 2144.6591,
                           'end': 2179.3182,
                           'value': 't'}
                         ],
              'value': 'wait'}
             ],
 'value': 'white'
 },
 ...
```

Nous avons créé également une autre méthode *toTextgrid* qui permet d'aligner l'audio et la transcription dans fichier TextGrid de Praat. Nous avons également fait une version statique de cette méthode *s_toTextgrid*. L'utilisateur à le choix d'avoir un alignement au niveau des mots seulement, des mots et des syllabes ou des mots, des syllabes et des phonèmes. La création des fichiers TextGrid nécessite que les timecodes soient parsés. Un exemple de fichier TextGrid est disponible à la page suivante. Les fichiers TextGrid portent automatiquement le même nom que les fichiers WAV.

Nous avons créé également deux méthodes statiques *updateDuration* et *updateTimecodeSpeed* qui permettent de recalculer la durée du fichier WAV et de mettre les *timecodes* à jour si l'utilisateur modifie la vitesse des fichiers WAV, comme nous l'avons fait nous-même avec SOX. Ainsi, lorsque nous avons modifié la vitesse des fichiers WAV, nous nous sommes servi de ces méthodes afin d'avoir des *timecodes* corrects.

8.4 Classe Speaker

Cette classe ne dispose d'aucune méthode particulière. Elle permet simplement de stocker dans les attributs de l'objet des informations sur le locuteur : nom, genre et nationalité.

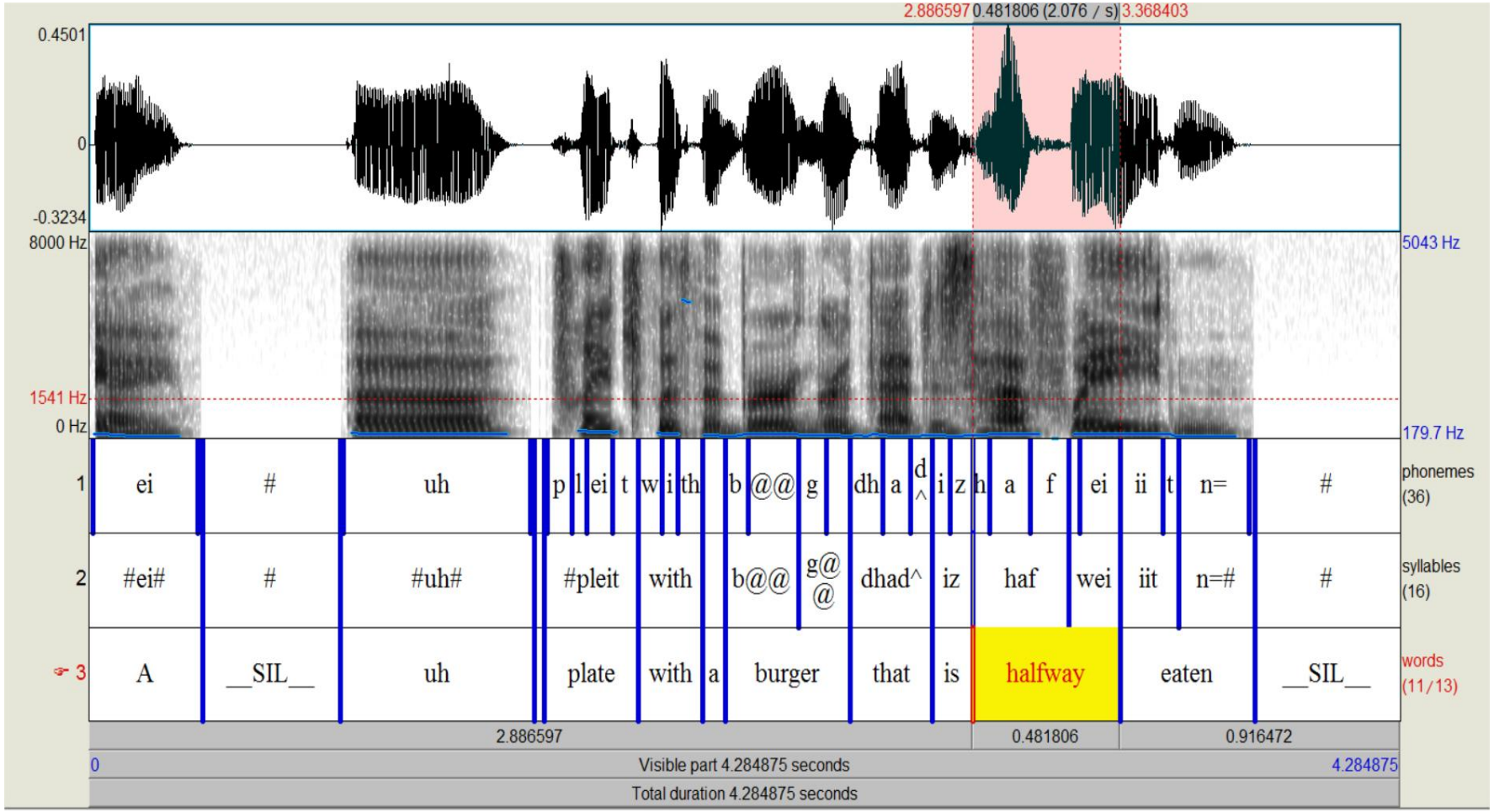


Figure 28 Version TextGrid Praat de la légende n°784361 de l'image n°1425

9 Résultats TF-IDF sur le corpus MBOSHI

Nous n'avons reproduit ici que les scores TF-IDF pour les fichiers WAV dont au moins un cluster contenait le nom de l'objet figuré dans l'image, ceux-ci sont surlignés en vert. En rouge figurent les clusters que nous avons classés comme étant non pertinents par observation de la matrice (page 96)

Top clusters in WAV file #0

Cluster: 25, TF-IDF: 0.32497
Cluster: 65, TF-IDF: 0.21664
Cluster: 100, TF-IDF: 0.21664
Cluster: 52, TF-IDF: 0.21664
Cluster: 54, TF-IDF: 0.0921
Cluster: 73, TF-IDF: 0.0921
Cluster: 39, TF-IDF: 0.0921
Cluster: 66, TF-IDF: 0.0921
Cluster: 67, TF-IDF: 0.0921
Cluster: 53, TF-IDF: 0.0921
Cluster: 26, TF-IDF: 0.0806
Cluster: 14, TF-IDF: 0.07167
Cluster: 16, TF-IDF: 0.06438
Cluster: 6, TF-IDF: 0.01932

Top clusters in WAV file #13

Cluster: 35, TF-IDF: 0.67701
Cluster: 40, TF-IDF: 0.28782
Cluster: 36, TF-IDF: 0.28782
Cluster: 26, TF-IDF: 0.25186
Cluster: 6, TF-IDF: 0.02587

Top clusters in WAV file #15

Cluster: 47, TF-IDF: 0.71264
Cluster: 62, TF-IDF: 0.28506
Cluster: 13, TF-IDF: 0.18861
Cluster: 44, TF-IDF: 0.15319
Cluster: 48, TF-IDF: 0.12119
Cluster: 12, TF-IDF: 0.12119
Cluster: 23, TF-IDF: 0.12119
Cluster: 61, TF-IDF: 0.12119
Cluster: 63, TF-IDF: 0.12119
Cluster: 1, TF-IDF: 0.0943
Cluster: 6, TF-IDF: 0.00726

Top clusters in WAV file #16

Cluster: 92, TF-IDF: 0.33851

Cluster: 85, TF-IDF: 0.16925
Cluster: 14, TF-IDF: 0.16798
Cluster: 102, TF-IDF: 0.14391
Cluster: 13, TF-IDF: 0.11198
Cluster: 3, TF-IDF: 0.08261
Cluster: 70, TF-IDF: 0.07196
Cluster: 90, TF-IDF: 0.07196
Cluster: 58, TF-IDF: 0.07196
Cluster: 68, TF-IDF: 0.06297
Cluster: 18, TF-IDF: 0.0413
Cluster: 6, TF-IDF: 0.02587

Top clusters in WAV file #23
Cluster: 44, TF-IDF: 0.4851
Cluster: 9, TF-IDF: 0.36107
Cluster: 10, TF-IDF: 0.26865
Cluster: 30, TF-IDF: 0.2389
Cluster: 104, TF-IDF: 0.15351
Cluster: 11, TF-IDF: 0.15351
Cluster: 6, TF-IDF: 0.0092

Top clusters in WAV file #28
Cluster: 4, TF-IDF: 0.5284
Cluster: 22, TF-IDF: 0.5284
Cluster: 23, TF-IDF: 0.22464
Cluster: 71, TF-IDF: 0.1321
Cluster: 99, TF-IDF: 0.1321
Cluster: 30, TF-IDF: 0.0874
Cluster: 79, TF-IDF: 0.05616
Cluster: 5, TF-IDF: 0.05616
Cluster: 7, TF-IDF: 0.05616
Cluster: 66, TF-IDF: 0.05616
Cluster: 70, TF-IDF: 0.05616
Cluster: 10, TF-IDF: 0.04914
Cluster: 1, TF-IDF: 0.0437
Cluster: 16, TF-IDF: 0.03925
Cluster: 6, TF-IDF: 0.01178

Top clusters in WAV file #3
Cluster: 3, TF-IDF: 0.31305
Cluster: 75, TF-IDF: 0.14253
Cluster: 76, TF-IDF: 0.14253
Cluster: 74, TF-IDF: 0.14253
Cluster: 50, TF-IDF: 0.06059
Cluster: 32, TF-IDF: 0.06059
Cluster: 31, TF-IDF: 0.06059
Cluster: 58, TF-IDF: 0.06059
Cluster: 6, TF-IDF: 0.0345

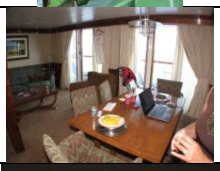
Top clusters in WAV file #5

Cluster: 94, TF-IDF: 0.23548
Cluster: 95, TF-IDF: 0.23548
Cluster: 87, TF-IDF: 0.17521
Cluster: 55, TF-IDF: 0.15581
Cluster: 3, TF-IDF: 0.11494
Cluster: 64, TF-IDF: 0.10011
Cluster: 28, TF-IDF: 0.10011
Cluster: 43, TF-IDF: 0.10011
Cluster: 40, TF-IDF: 0.10011
Cluster: 6, TF-IDF: 0.027

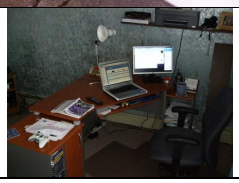
Top clusters in WAV file #7

Cluster: 33, TF-IDF: 0.20933
Cluster: 34, TF-IDF: 0.20933
Cluster: 24, TF-IDF: 0.20933
Cluster: 44, TF-IDF: 0.1323
Cluster: 6, TF-IDF: 0.0439

10 Licences d'utilisation des images de MSCOCO

Image	Adresse	Auteur	Licence
	https://www.flickr.com/photos/sheila_steele/46176722/ [consulté le 24 mai 2017]	Sheila Steele	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/wonderlane/2299974764/ [consulté le 24 mai 2017]	Wonderlane	CC BY 2.0[52]
	https://www.flickr.com/photos/mckln/3504254524/ [consulté le 24 mai 2017]	David Woo	CC BY-ND 2.0[53]
	https://www.flickr.com/photos/40295335@N00/4479479414/ [consulté le 24 mai 2017]	Joel Abroad	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/rmvandy/5394114231/ [consulté le 24 mai 2017]	Bob	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/ahmadnawawi/5613889435/ [consulté le 24 mai 2017]	Ahmad Nawawi	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/civellod/6831766923/ [consulté le 24 mai 2017]	Daniele Civello	CC BY-NC 2.0[54]

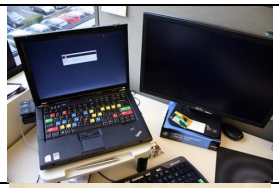
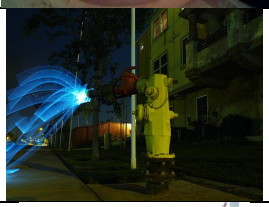
[51] https://creativecommons.org/licenses/by-nc-sa/2.0/ [consulté le 24 mai 2017]
[52] https://creativecommons.org/licenses/by/2.0/ [consulté le 24 mai 2017]
[53] https://creativecommons.org/licenses/by-nd/2.0/ [consulté le 24 mai 2017]
[54] https://creativecommons.org/licenses/by-nc/2.0/ [consulté le 24 mai 2017]

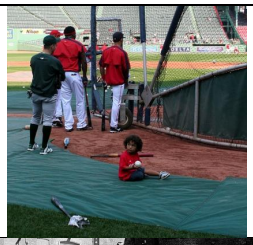
	https://www.flickr.com/photos/tim_proffitt_white/7017582105/ [consulté le 24 mai 2017]	Tim Proffitt-White	CC BY-NC-ND 2.0[55]
	https://www.flickr.com/photos/minutemade/7680520840/ [consulté le 24 mai 2017]	Square Gato	CC BY-SA 2.0[56]
	https://www.flickr.com/photos/cumidanciki/8026451116/ [consulté le 24 mai 2017]	@ccfoodtravel	CC BY 2.0[57]
	https://www.flickr.com/photos/danimarques/8297056781/ [consulté le 24 mai 2017]	Daniela Marques	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/infomatique/2796261330/ [consulté le 24 mai 2017]	William Murphy	CC BY 2.0[57]
	https://www.flickr.com/photos/mtsofan/9461789076/ [consulté le 24 mai 2017]	John	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/photophonic/388877361/ [consulté le 24 mai 2017]	Mike Baehr	CC BY-NC-SA 2.0[51]
	https://www.flickr.com/photos/tachyondecay/2366167297/ [consulté le 24 mai 2017]	Ben Babock	CC BY 2.0[57]

[55] https://creativecommons.org/licenses/by-nc-nd/2.0/ [consulté le 24 mai 2017]

[56] https://creativecommons.org/licenses/by-sa/2.0/ [consulté le 24 mai 2017]

[57] https://creativecommons.org/licenses/by/2.0/ [consulté le 24 mai 2017]

	https://www.flickr.com/photos/soulsoap/3154457418/ [consulté le 24 mai 2017]	Matt Cummings	CC BY-NC-ND 2.0[55]
	https://www.flickr.com/photos/lookcatalog/8117025068/ [consulté le 24 mai 2017]	Thoughtcatalog.com	CC BY-NC-ND 2.0[55]
	https://www.flickr.com/photos/freakapotimus/2791132570/ [consulté le 24 mai 2017]	Reed	CC BY 2.0[57]
	https://www.flickr.com/photos/tr0ubleshh00ter/5087434568/ [consulté le 24 mai 2017]	Tr0ubleSh00ter	CC BY 2.0[57]
	https://www.flickr.com/photos/sudama/3893618/ [consulté le 24 mai 2017]	Adam Rice	CC BY-NC 2.0[54]
	https://www.flickr.com/photos/chrisostermann/1283521760/ [consulté le 24 mai 2017]	Chris Ostermann	CC BY 2.0[57]
	https://www.flickr.com/photos/mikek/162225436/ [consulté le 24 mai 2017]	Mike Kuniavsky	CC BY-NC-SA 2.0[51]

	https://www.flickr.com/photos/aerorace re/1470921244/ [consulté le 24 mai 2017]	Ethan Bagley	CC BY-NC-ND 2.0[55]
	https://www.flickr.com/photos/demond_henderson/5839665221/ [consulté le 24 mai 2017]	Demond Henderson	CC BY-NC-ND 2.0[55]

Mots clefs : TAL, corpus multimodal, découverte non supervisée, langues en danger, documentation, lexique

RÉSUMÉ

De nombreuses langues disparaissent tous les ans et ce à un rythme jamais atteint auparavant. Les linguistes de terrain manquent de temps et de moyens afin de pouvoir toutes les documenter et décrire avant qu'elles ne disparaissent à jamais. L'objectif de notre travail est donc de les aider dans leur tâche en facilitant le traitement des données. Nous proposons dans ce mémoire des méthodes d'extraction non supervisées de lexique à partir de corpus multimodaux incluant des signaux de parole et des images. Nous proposons également une méthode issue de la recherche d'information afin d'émettre des hypothèses de signification sur les éléments lexicaux découverts. Ce mémoire présente en premier lieu la constitution d'un corpus multimodal parole-image de grande taille. Ce corpus simulant une langue en danger permet ainsi de tester les approches computationnelles de découverte non supervisée de lexique. Dans une seconde partie, nous appliquons un algorithme de découverte non supervisée de lexique utilisant de l'alignement dynamique temporel segmental (S-DTW) sur un corpus multimodal synthétique de grande taille ainsi que sur un corpus multimodal d'une vraie langue en danger, le Mboshi.

Keywords : NLP, multimodal corpus, unsupervised term discovery, UTD, endangered languages, documentation

ABSTRACT

Many languages are on the brink of extinction and many disappear each and every year at a rate never seen before. Field linguists lack the time and the means to document and describe all of them before they die out. The goal of our work is to help them in their task, make it easier and speed up the data processing and annotation tasks. In this dissertation, we propose methods to use an unsupervised term discovery (UTD) system to extract lexicon from multimodal corpora consisting of speech and images. We also propose a method using information retrieval techniques to hypothesise the meaning of the discovered lexical items. In the first place, this dissertation presents the creation of a large multimodal corpus which includes speech and images. This corpus simulating that of an endangered language will allow us evaluate the performances of an unsupervised term discovery system. In the second place, we apply an unsupervised term discovery system based on segmental dynamic time warping (S-DTW) to a large synthetic multimodal corpus and also to the multimodal corpus of a real endangered language called Mboshi, spoken in Congo-Brazzaville.