Synthèse incrémentale de la parole à partir du texte

Maël Pouget

UNIVERSITÉ GRENOBLE ALPES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE-ALPES

Spécialité : **Signal, Image, Parole, Télécom (SIPT)**

Présentée par
Maël POUGET

Thèse dirigée par **Gérard BAILLY** et
co-encadrée par **Thomas Hueber**
préparée au sein du
Département Parole et Cognition (DPC) du GIPSA-Lab, Grenobe
dans **l'École doctorale Électonique, Électrotechnique, Automatique et Traitement du Signal (EEATS)**

Synthèse incrémentale de la parole à partir du texte.

Thèse soutenue publiquement le **23 juin 2017**,
devant le jury composé de :

M. Christophe D'ALESSANDRO
Directeur de recherche CNRS, Institut Jean le Rond d'Alembert, Paris
M. Frédéric BÉCHET
Professeur, Université d'Aix-Marseille, France
M. Laurent BESACIER
Professeur, Université Grenoble-Alpes, Grenoble, Président du jury
M. Damien LOLIVE
Maitre de Conférences, Université Rennes 1, Rennes
M. Gérard BAILLY
Directeur de Recherche CNRS, Gipsa-Lab, Grenoble, Directeur de thèse
M. Thomas HUEBER
Chargé de Recherche CNRS, Gipsa-Lab, Grenoble, Encadrant de thèse

UNIVERSITÉ DE GRENOBLE-ALPES

ÉCOLE DOCTORALE EEATS

Électronique, Électrotechnique, Automatique, Traitement du Signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université Grenoble-Alpes

Mention : Signal Image Parole Télécom

Présentée et soutenue par

Maël POUGET

Synthèse incrémentale de la parole à partir du texte.

Thèse dirigée par Gérard BAILLY

et co-encadrée par Thomas HUEBER

préparée au GIPSA-Lab

soutenue le 23 Juin 2017

Jury :

Rapporteurs :	Christophe D'ALESSANDRO	-	CNRS, Institut Jean le Rond d'Alembert
	Frédéric BÉCHET	-	Université d'Aix-Marseille
Directeur :	Gérard BAILLY	-	CNRS, Gipsa-Lab
Encadrant :	Thomas HUEBER	-	CNRS, Gipsa-Lab
Président du jury :	Laurent BESACIER	-	Université Grenoble-Alpes
Examinateur :	Damien LOLIVE	-	Univeristé Rennes 1

Résumé — Ce travail de thèse porte sur un nouveau paradigme pour la synthèse de la parole à partir du texte, à savoir la synthèse incrémentale. L'objectif est de délivrer la parole de synthèse au fur et à mesure de la saisie du texte par l'utilisateur, contrairement aux systèmes classiques pour lesquels la synthèse est déclenchée après la saisie d'une ou plusieurs phrases. L'application principale visée est l'aide aux personnes présentant un trouble sévère de la communication orale, et communiquant principalement à l'aide d'un synthétiseur vocal. Un synthétiseur vocal incrémental permettrait de fluidifier une conversation en limitant le temps que passe l'interlocuteur à attendre la fin de la saisie de la phrase à synthétiser. Un des défi que pose ce paradigme est la synthèse d'un mot ou d'un groupe de mots avec une qualité segmentale et prosodique acceptable alors que la phrase qui le contient n'est que partiellement connue au moment de la synthèse. Pour ce faire, nous proposons différentes adaptations des deux principaux modules d'un système de synthèse de parole à partir du texte : le module de traitement automatique de la langue naturelle (TAL) et le module de synthèse sonore.

Pour le TAL en synthèse incrémentale, nous nous sommes intéressé à l'analyse morpho-syntaxique, qui est une étape décisive pour la phonétisation et la détermination de la prosodie cible. Nous décrivons un algorithme d'analyse morpho-syntaxique dit "à latence adaptative". Ce dernier estime en ligne si une classe lexicale (estimée à l'aide d'un analyseur morpho-syntaxique standard basé sur l'approche *n-gram*), est susceptible de changer après l'ajout d'un ou plusieurs mots par l'utilisateur. Si la classe est jugée instable, alors la synthèse sonore est retardée, dans le cas contraire, elle peut s'effectuer sans risque a priori de dégrader la qualité segmentale et suprasegmentale. Cet algorithme exploite un ensemble d'arbres de décision binaire dont les paramètres sont estimés par apprentissage automatique sur un large corpus de texte. Cette méthode nous permet de réaliser un étiquetage morpho-syntaxique incrémental avec une précision de 92,5% pour une latence moyenne de 1,4 mots.

Pour la synthèse sonore, nous nous plaçons dans le cadre de la synthèse paramétrique statistique, basée sur les modèles de Markov cachés (Hidden Markov Models, HMM). Nous proposons une méthode de construction de la voix de synthèse (estimation des paramètres de modèles HMM) prenant en compte une éventuelle incertitude sur la valeur de certains descripteurs contextuels qui ne peuvent pas être calculés en synthèse incrémentale (c'est-à-dire ceux qui portent sur les mots qui ne sont pas encore saisis au moment de la synthèse). Nous comparons la méthode proposée à deux autres stratégies décrites dans la littérature. Les résultats des évaluations objectives et perceptives montrent l'intérêt de la méthode proposée pour la langue française.

Enfin, nous décrivons un prototype complet qui combine les deux méthodes proposées pour le TAL et la synthèse par HMM incrémentale. Une évaluation perceptive de la pertinence et de la qualité des groupes de mots synthétisés au fur et à mesure de la saisie montre que notre système réalise un compromis acceptable entre réactivité (minimisation du temps entre la saisie d'un mot et sa synthèse) et qualité (segmentale et prosodique) de la parole de synthèse.

Abstract — In this thesis, we investigate a new paradigm for text-to-speech synthesis (TTS) allowing to deliver synthetic speech while the text is being inputted : incremental text-to-speech synthesis. Contrary to conventional TTS systems, that trigger the synthesis after a whole sentence has been typed down, incremental TTS devices deliver speech in a "piecemeal" fashion (i.e. word after word) while aiming at preserving the speech quality achievable by conventional TTS systems. By reducing the waiting time between two speech outputs while maintaining a good speech quality, such a system should improve the quality of the interaction for speech-impaired people using TTS devices to express themselves. The main challenge brought by incremental TTS is the synthesis of a word, or of a group of words, with the same segmental and supra-segmental quality as conventional TTS, but without knowing the end of the sentence to be synthesized. In this thesis, we propose to adapt the two main modules (natural language processing and speech synthesis) of a TTS system to the incremental paradigm.

For the natural language processing module, we focused on part-of-speech tagging, which is a key step for phonetization and prosody generation. We propose an "adaptive latency algorithm" for part-of-speech tagging, that estimates if the inferred part-of-speech for a given word (based on the *n-gram* approach) is likely to change when adding one or several words. If the Part-of-speech is considered as likely to change, the synthesis of the word is delayed. In the other case, the word may be synthesized without risking to alter the segmental or supra-segmental quality of the synthetic speech. The proposed method is based on a set of binary decision trees trained over a large corpus of text. We achieve 92.5% precision for the incremental part-of-speech tagging task and a mean delay of 1.4 words.

For the speech synthesis module, in the context of HMM-based speech synthesis, we propose a training method that takes into account the uncertainty about contextual features that cannot be computed at synthesis time (namely, contextual features related to the following words). We compare the proposed method to other strategies (baselines) described in the literature. Objective and subjective evaluation show that the proposed method outperforms the baselines for French.

Finally, we describe a prototype developed during this thesis implementing the proposed solution for incremental part-of-speech tagging and speech synthesis. A perceptive evaluation of the word grouping derived from the proposed adaptive latency algorithm as well as the segmental quality of the synthetic speech tends to show that our system reaches a good trade-off between reactivity (minimizing the waiting time between the input and the synthesis of a word) and speech quality (both at segmental and supra-segmental levels).

Remerciements

On peut s'entendre dire qu'une thèse, c'est 3 ans de relative solitude, en tête-à-tête avec son sujet, ses expériences et ses rédactions. Et bien aujourd'hui, après 3 ans (et quelques mois) de travail, je sais que c'est faux. Je souhaite exprimer mes plus sincères remerciements à mes directeurs de thèse, Gérard Bailly et Thomas Hueber, dont le travail et le soutien m'ont permis de maintenir le cap, que ce soit pour les choix théoriques un peu compliqués ou pour les péripéties ô combien éprouvantes de la rédaction.

Thomas, tu as été, en ce qui me concerne, un formidable directeur de thèse, principalement via ta passion pour mon sujet qui se ressentait à travers ton encadrement. Te voir autant impliqué et motivé par toutes les petites contributions que j'apportais m'a vraiment aidé à moi-même rester motivé.

Gérard, ton expérience incroyablement riche dans le domaine de la parole m'a également beaucoup apporté. Chaque discussion à propos de la direction dans laquelle mener ma thèse se retrouvait ponctuée d'un *"attend, untel a écrit ça dans un papier, je te le retrouve"*. Se savoir encadré par un chercheur armé d'une telle culture scientifique a été un honneur.

Je tiens également à remercier les membres de mon jury de thèse, messieurs D'Alessandro, Béchet, Lolive et Besacier, qui ont lu mon travail avec une grande attention et qui ont fait de ma soutenance un moment plaisant dont je garderai un agréable souvenir.

Enfin, je souhaite remercier Sylvain Gerber, pour ses connaissances en statistiques qui m'ont permis d'éviter de tomber dans le *"tout le monde fait ça, alors je vais le faire aussi"*. Pour la réalisation de mon prototype, je souhaite formuler un grand merci à Olha Nahorna, qui a beaucoup travaillé sur la réalisation de la version en ligne, et à Anne Fradin, dont l'aide a permis de rendre le tout fonctionnel.

Que ce soit au labo autant qu'en dehors, il y a tant de personnes que je souhaite remercier que j'espère ne pas en oublier. Merci aux doctorants, aux stagiaires et aux RH qui ont toujours été source de rire, de discussions plus ou moins sérieuses, ou de parties endiablées de belote : Diandra, Adela, Guillaume, Raphaël, Gaël (B306 RPZ), Rémi, Sophie, Alexandre, Jean-Francois, Thibaut, Firas, Polina, Marie-Lou, Thomas, Aline, Grégoire, Elsa, Laure, Marion, Sandra, Cindy et Cécilia.

Merci également aux chercheurs du GIPSA-Lab pour les connaissances qu'ils ont partagées avec moi durant de passionnantes discussions, et notamment à Pascal, qui, lors de mon entrée au laboratoire était un des professeurs qui m'avaient amené à faire du traitement de la parole, et qui, maintenant que j'en sors, est un ami.

Enfin, merci aux amis, aux collocs et à ma famille qui m'ont soutenu et apporté tout plein d'autres choses durant ces quelques années, et qui surtout sont venus d'ici et d'ailleurs pour assister à ce point d'orgue qu'est la soutenance (et qui m'ont offert un amphi d'école d'ingénieur plus rempli que la moyenne annuelle). En particulier, pour leur soutien inconditionnel, merci à Julien, Jimmy, Laurent, Florian et Isie.

Table des matières

Table des figures

Liste des tableaux

Table des sigles et acronymes

API	Application Programming Interface
AR	Auto-Régressif
CRF	Conditionnal Random Fields
HMM	Hidden Markov Model
HNM	Harmonique Plus Noise
HSMM	Hidden Semi-Markov Model
CD-HSMM	Context-Dependant Hidden Semi-Markov Model
CI-HSMM	Context-Independant Hidden Semi-Markov Model
HTS	HMM-based Speech Synthesis System (H Triple S)
LPC	Linear Predictive Coding
LSF	Line Spectral Frequency
LSP	Line Spectrum Pairs
LSTM	Long Short-Term Memory
MGC	Mel Generalized Cepstrum
MLPG	Maximum Likelihood Parameter Generation
MLSA	Mel-Log Spectrum Approximation
MOS	Mean Opinion Score
MSE	Mean Square Error
TAL	Traitement Automatique de la Langue naturelle
TTS	Text-To-Speech

Chapitre 1

Introduction

Sommaire

1.1 Contexte et motivations

Un système de synthèse de parole à partir du texte (TTS : Text-To-Speech) est un système capable de générer un signal audio de parole à partir de n'importe quel texte. Les systèmes TTS ont aujourd'hui atteint une qualité suffisante pour être déployés dans des applications tout public. On les retrouve ainsi dans de nombreuses applications telles que l'aide à la navigation (GPS), les serveurs vocaux téléphoniques, la diffusion de messages dans les lieux publics, la robotique humanoïde, les assistants virtuels des *smartphones* (Siri, Cortana, etc.), l'industrie du divertissement (jeu vidéo, etc) ou encore l'assistance aux personnes atteintes de déficiences visuelles (via par exemple le *voicemail* ou la lecture automatique de pages Internet etc.). Par ailleurs, les systèmes TTS, éventuellement couplés à des interfaces d'aide à la saisie de texte (pictogramme, saisie prédictive de texte, etc.), constituent pour certaines personnes un système complet de suppléance vocale. C'est notamment le cas de personnes atteintes de maladies neurodégénératives telles que la SLA (la Sclérose Latérale Amyotrophique, 1000 cas diagnostiqués par an en France [1]) ou de cancers des voies aérodigestives supérieures (16 000 nouveaux par an cas en France [2]) notamment du larynx (environ 3 100 nouveaux cas par an en France) [3].

Cependant, le paradigme classique en synthèse de parole à partir du texte est la synthèse de phrases (ou de paragraphes). L'analyse du texte et la synthèse sonore sont déclenchées à chaque fois que l'utilisateur a terminé la saisie d'une phrase "complète", indiquée classiquement par la présence d'un marqueur de fin de phrase (tel que le point). Comme nous le détaillerons ultérieurement, la connaissance des limites de début et de fin de phrase est importante pour son analyse linguistique, notamment pour déterminer la classe lexicale des mots qui la constitue et sa structure syntaxique, c'est-à-dire les relations d'ordre et de dominance entre chacun des mots qui la constitue. Une analyse linguistique précise est primordiale pour que la parole de synthèse ait une bonne qualité segmentale – c'est-à-dire une restitution correcte de la chaîne phonétique cible – et suprasegmentale, c'est-à-dire un contenu prosodique naturel [4].

Dans ce travail de thèse, nous faisons l'hypothèse que ce paradigme de synthèse "phrase-à-phrase" est mal adapté à une situation où la synthèse TTS est utilisée comme système de suppléance vocale par une personne en situation de handicap. En effet, ce paradigme est susceptible d'introduire une latence importante (et proportionnelle à la longueur de la phrase) dans la communication entre deux interlocuteurs. Il peut être à l'origine d'une certaine frustration pour le destinataire de la communication qui est contraint d'attendre la fin de la saisie de chaque phrase pour comprendre et réagir aux propos de son interlocuteur comme

1. selon `http://www.arsla.org/la-sla-en-chiffres/`
2. Les données sur les cancers des voies aérodigestives supérieures sont disponibles sur le site `http://www.arcagy.org/infocancer/localisations/voies-aeriennes/cancers-du-larynx/maladie/avant-propos.html`
3. dans le cas du cancer du larynx, la synthèse TTS peut être utile en attente de la mise en place d'une voix de substitution telle que la voix œsophagienne ou tracheo-œsophagienne, ou de la prise en main d'un électro-larynx
4. la prosodie est ici définie comme l'ensemble des variations de hauteur, de niveau sonore et de longueurs syllabiques, tel que proposé par (Di Cristo 2000)

pour l'utilisateur du TTS qui peut être tenté de simplifier son discours pour limiter cette attente, et maintenir ainsi une certaine fluidité dans l'interaction. Cette situation est illustrée à la figure 1.1.

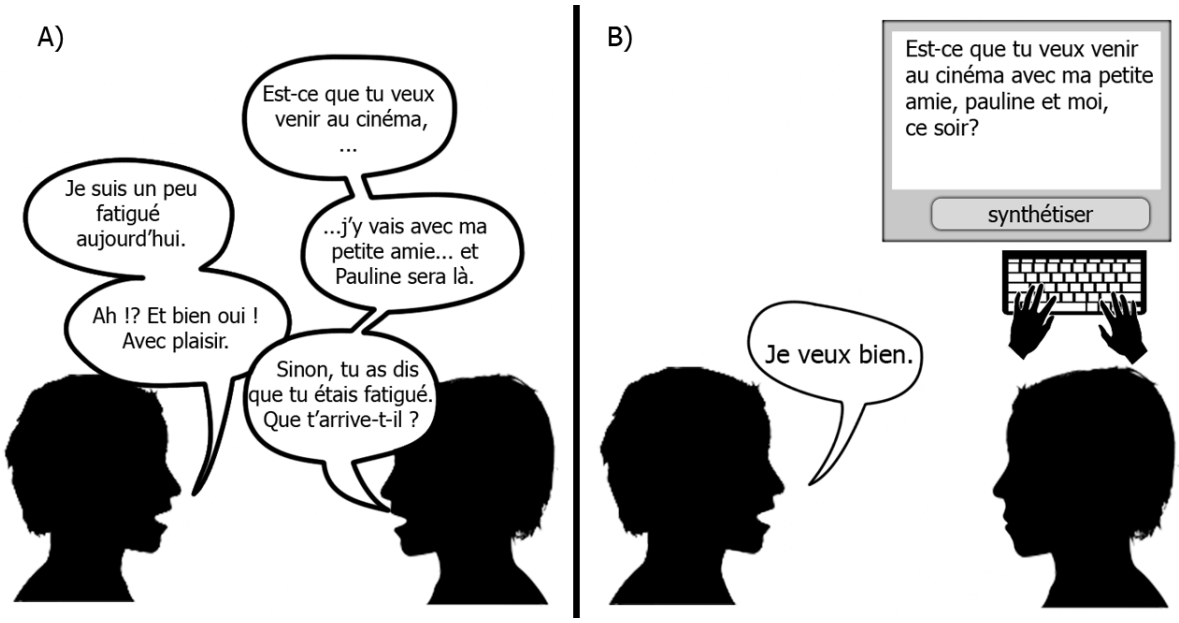


FIGURE 1.1 – Illustration du manque d'interactivité dans une conversation orale lorsqu'un des participants utilise un système TTS. À gauche, une interaction fluide pendant laquelle les deux interlocuteurs s'interrompent, exploitent leurs hésitations mutuelles pour prendre leur tour de parole, tentent d'anticiper ce que l'autre va dire, etc. accélérant ainsi l'échange d'information. A droite, la même interaction lorsqu'un des interlocuteurs utilise un système TTS et un paradigme de synthèse "par phrase" pour communiquer. Le destinataire est contraint d'attendre la fin de la saisie de chaque phrase par l'émetteur pour lui répondre. La fluidité et donc la qualité de l'interaction est susceptible d'être considérablement dégradée.

Pour pallier ce problème, certains utilisateurs préfèrent déclencher la synthèse vocale après la saisie de chaque phonème ou de chaque mot (c'est notamment une option des systèmes TTS à destination des personnes en situation de handicap, tel que le *Lightwriter SL40* de la société Toby Churchill). Cette stratégie diminue évidemment la latence entre la saisie du texte et sa synthèse et améliore donc la qualité de l'interaction (l'auditeur peut "anticiper" et "réagir" au fur et à mesure de l'écoute de la parole de synthèse). En revanche, cela se fait au détriment de la qualité de la parole de synthèse, dont la génération ne s'appuie que sur la connaissance du mot à synthétiser, indépendamment de son contexte linguistique (sa position dans la phrase, sa fonction grammaticale, etc.).

Dans ce travail de thèse, nous abordons ce problème à l'aide du paradigme dit de synthèse incrémentale. La synthèse incrémentale vise à délivrer, au fur et à mesure de la saisie du texte par l'utilisateur, une parole de synthèse à la qualité (segmentale et suprasegmentale) proche de la qualité obtenue à l'aide d'un paradigme de synthèse "phrase-à-phrase". En synthèse incrémentale, la parole de synthèse accompagne la saisie. Dans le système envisagé, la synthèse

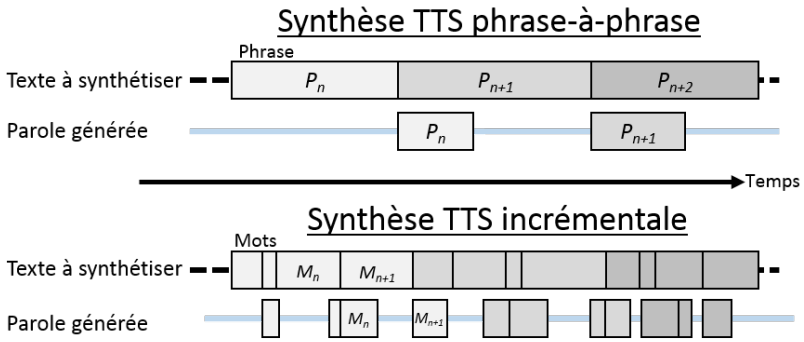


FIGURE 1.2 – Illustration du paradigme de synthèse TTS incrémentale en comparaison avec la synthèse TTS classique "phrase-à-phrase". En synthèse incrémentale, la parole artificielle accompagne la saisie du texte.

est au mieux déclenchée après la saisie d'un mot et, comme nous le verrons ultérieurement, peut aller jusqu'à la synthèse d'un groupe de plusieurs mots. Le paradigme de la synthèse incrémentale est illustré à la figure 1.2. Il y est comparé à la synthèse classique phrase-à-phrase.

La synthèse incrémentale cherche ainsi un compromis entre "qualité" de la parole de synthèse, et "réactivité" du système TTS. Comme illustré à la Figure 1.3, nous cherchons un intermédiaire entre une synthèse mot-à-mot de faible qualité (notamment prosodique) mais très réactive (la latence est constante et égale un mot), et une synthèse phrase-à-phrase de très haute qualité, mais peu réactive (d'une latence proportionnelle à la longueur de la phrase). Dans le cadre de l'aide au handicap, nous faisons l'hypothèse qu'un tel système permettra une interaction conversationnelle plus fluide qu'un système TTS classique, mais sans pour autant sacrifier la qualité de la parole artificielle.

1.2 Applications de la synthèse vocale à partir du texte

Si la recherche en synthèse vocale est toujours très active, de nombreux systèmes commerciaux sont aujourd'hui disponibles tels que les systèmes de *Acapela*, *Voxygen*, *Nuance*, *Google TTS*, *Siri Text-to-speech* (*Apple*), etc., et sont déployés dans différentes applications comme les assistants virtuels, où ils constituent un des bouts de la chaîne d'un système de dialogue, tel qu'illustré à la figure 1.4.

La synthèse vocale est également un moyen de communication pour les personnes en situation de handicap. La synthèse TTS permet notamment le développement d'applications de lecture de texte et d'audio-description pour les personnes mal-voyantes ou non-voyantes telles que *VoiceOver* (par *Apple*), *JAWS* (*Windows*) ou *TalkBack* (*Android*). La synthèse TTS est également utilisée comme outil de suppléance vocale par des personnes en situation de handicap. Les sociétés Aria et Leblat proposent par exemple (Figure 1.5) des systèmes de synthèse de parole portatifs dont la saisie se fait phonème par phonème (voire par syllabes).

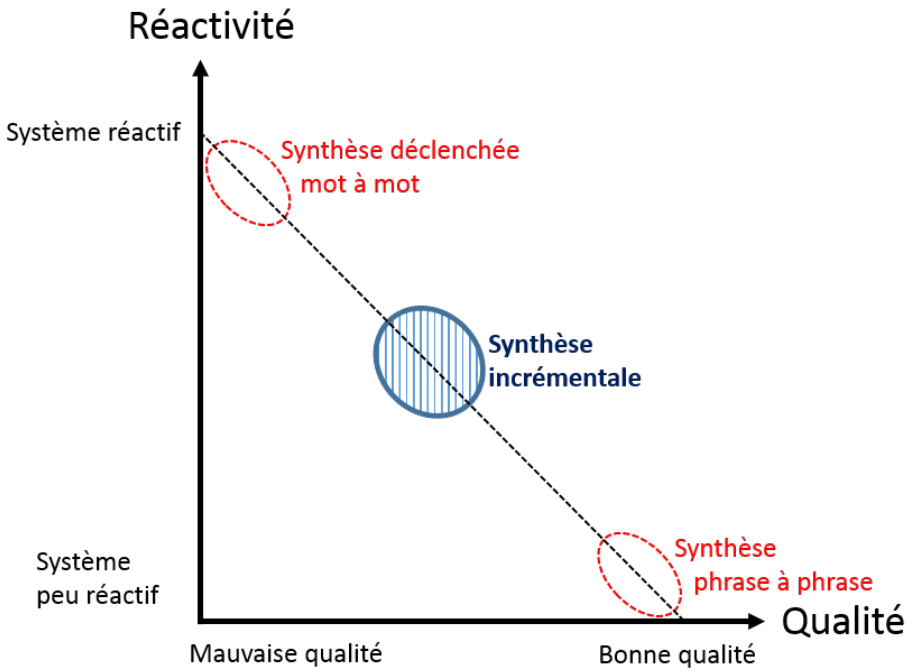


FIGURE 1.3 – Synthèse mot à mot, synthèse incrémentale, et synthèse phrase-à-phrase : compromis entre réactivité du système et qualité de la parole artificielle.

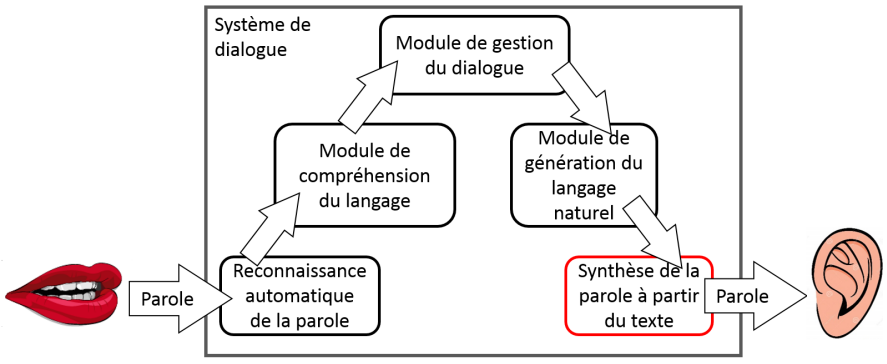


FIGURE 1.4 – Illustration des différents modules constituant un système de dialogue. La synthèse vocale est ici utilisée en bout de chaîne pour délivrer la parole de synthèse issues du module de génération automatique du langage naturel. Figure inspirée de (BAUMANN 2013).

(a) Système *Synthé 5* de la société Aria synthétisant de la parole saisie phonème par phonème.

(b) Le Leblaphone, développé par la société Leblat, permet de synthétiser de la parole en saisissant des syllabes.

FIGURE 1.5 – Systèmes TTS commerciaux fonctionnant par saisie de phonèmes ou de syllabes.

Ces systèmes, présentant l'avantage d'être facilement transportables, synthétisent de la parole concaténant des phonèmes ou des syllabes pré-enregistrés. Ces systèmes sont donc très réactifs (la parole est délivrée syllabe par syllabe pour le Leblatphone, Figure 1.5b). Cependant, la parole de synthèse qui en résulte est peu naturelle.

Dans le système *Lightwriter SL40* de la société Toby Churchill illustré à la Figure 1.6a, le synthétiseur peut vocaliser chaque mot de façon isolée dès sa saisie. Cette synthèse hors-contexte est susceptible d'aboutir, dans certains cas, à une phonétisation incorrecte et une intonation peu naturelle (montée et descente intonative pour chaque mot qui est considéré par le système comme une "phrase"). Dans ce système, l'utilisateur peut ensuite déclencher une synthèse classique de bonne qualité une fois la phrase complètement saisie.

Il existe également des systèmes, tel que le *Dynavox* de la société Toby Churchill (Figure 1.6b), pour lesquels l'utilisateur ne saisit pas directement le texte, mais le génère à l'aide d'une interface utilisant par exemple des pictogrammes. On citera notamment les travaux de (BLACHE et RAUZY 2007), sur la génération de langage naturel à partir d'icônes/pictogrammes dans le cadre du développement du logiciel *Plateforme de Communication Alternative*[5].

1.3 Qualité d'une interaction conversationnelle

Comme mentionné ci-avant, le paradigme de synthèse phrase-à-phrase introduit une latence importante entre les échanges des interlocuteurs. (RICHARDS 1973) décrit une conversation (face-à-face ou par l'intermédiaire d'un système de télécommunication) comme un échange

5. Ce type d'interface n'est cependant pas considéré dans le cadre de ce travail où nous nous restreignons à une entrée textuelle.

(a) *Lightwriter* développé par la société Toby Churchill. Ce système permet de synthétiser de la parole mot-à-mot, au fur et à mesure de la saisie. Lors de la fin de la saisie de la phrase, celle-ci est entièrement répétée avec, cette fois ci, une prosodie appropriée.

(b) *Dynavox*, développé par la société Toby Churchill. La saisie de texte peut se faire par l'intermédiaire de lettres, de mots, de pictogrammes ou en combinant ces différentes méthodes de saisie.

FIGURE 1.6 – Systèmes TTS commerciaux à destination des personnes en situation de handicap fonctionnant par saisie de mots ou de pictogrammes.

successif d'informations entre deux participants. Les participants prennent successivement les rôles de locuteur et d'auditeur, cette alternance des rôles permettant de créer une interaction entre eux. Or, il arrive que, durant l'interaction, les deux participants prennent simultanément le rôle de locuteur (on se trouve dans une situation de parole simultanée) ou d'auditeur (auquel cas, on se trouve dans une situation de silence mutuel). Sur la base de ces travaux, (GUÉGUIN 2006) identifie le délai entre la production de la parole et sa réception par l'interlocuteur comme une mesure possible de la qualité de l'interaction dans le cadre de communications téléphoniques longue distance. Elle montre notamment que dans le cadre d'une conversation téléphonique, l'interaction est fortement dégradée dès que ce délai dépasse 400ms. Bien que cette étude ait été réalisée dans un cadre différent de celui de cette thèse, ces résultats appuient l'hypothèse selon laquelle un délai trop important dans une interaction conversationnelle effectuée par l'intermédiaire d'un système TTS nuit grandement à l'interaction. Dans ce cas, le délai entre saisie et synthèse, égal au minimum à la longueur de la phrase à synthétiser, est susceptible d'être supérieur à 400ms.

Par ailleurs, la prosodie joue un rôle essentiel dans la parole, et de ce fait, dans une interaction conversationnelle. Elle joue un rôle primordial dans la structuration du discours et permet de véhiculer de l'information de haut niveau liée au sens telle que la mise en relief, mais aussi l'assertion, l'injonction, l'interrogation, etc. Aussi, une parole de synthèse générée mot-à-mot, et donc au contenu prosodique peu réaliste, est difficile (et "fatigante") à comprendre pour l'interlocuteur, qui n'a que très peu d'indices acoustiques lui permettant de distinguer les fins de phrase, les pauses syntaxiques, etc. Un des défis de la synthèse incrémentale est donc

de restituer de façon réactive (c'est-à-dire au fur et à mesure de la saisie du texte) une parole à la qualité segmentale et suprasegmentale (prosodique) la meilleure possible (proche de celle obtenue en synthèse phrase-à-phrase et meilleure que celle obtenue en synthèse mot-à-mot).

1.4 Fonctionnement général d'un système TTS

Classiquement, et comme nous pouvons le voir à la Figure 1.7, un synthétiseur de parole à partir du texte comporte deux modules principaux : un premier module dit de "Traitement Automatique de la Langue naturelle", que nous appellerons par la suite "module de TAL", et un module de "synthèse sonore", que nous appellerons par la suite "module de synthèse".

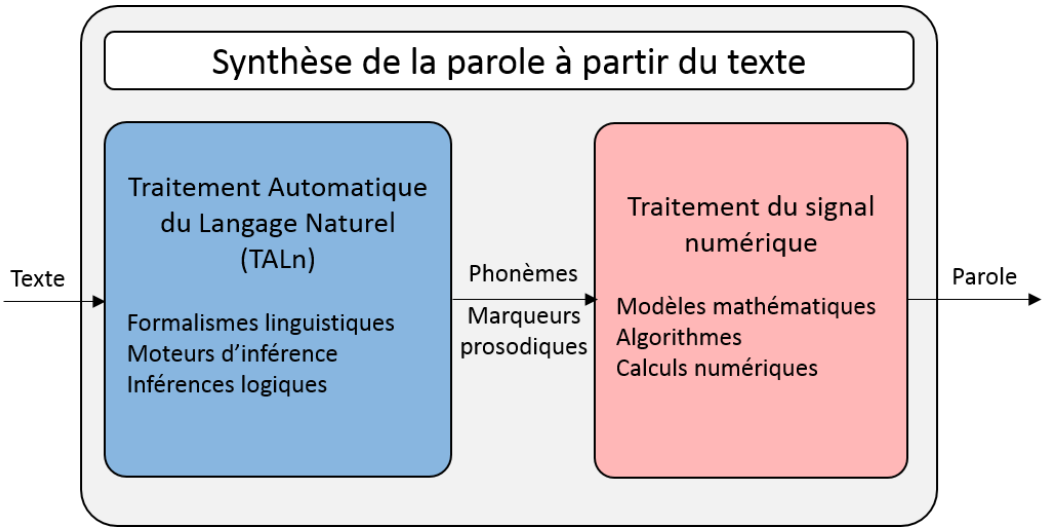


FIGURE 1.7 – Diagramme fonctionnel d'un système de synthèse *Text-to-speech* (TTS). Schéma inspiré de BOITE et al. 2000.

Le module de TAL analyse la phrase à synthétiser à différents niveaux linguistiques. Il en extrait une transcription phonétique [6] ainsi que sa structure syntaxique. Son rôle est de calculer un ensemble de descripteurs décrivant le contexte linguistique de chacun des phonèmes à synthétiser.

Le module de synthèse exploite ces différentes informations pour générer le signal audio de parole, à l'aide de différentes techniques, comme la synthèse par concaténation d'unités, ou la synthèse paramétrique statistique, qui est la technique privilégiée dans le cadre de ce travail.

Dans ce travail, nous proposons différentes adaptations de chacun de ces modules pour réaliser la synthèse TTS de façon incrémentale. Comme nous le détaillerons ultérieurement,

6. (VAISSIÈRE 2011) définit le phonème comme étant "la plus petite unité fonctionnelle d'un système phonologique. La fonction des phonèmes dans une langue est d'établir des oppositions entre les mots de son lexique. Si deux sons apparaissent exactement à la même position phonique et ne peuvent se substituer l'un à l'autre sans modifier la signification des mots, ou sans que le mot devienne méconnaissable, alors les deux sons sont des réalisations de deux phonèmes".

le problème que pose ce paradigme est l'analyse linguistique et la synthèse sonore d'un mot sans connaître son contexte complet, c'est-à-dire en s'appuyant uniquement sur les mots déjà saisis.

Glossaire et définitions

Nous proposons à présent de fixer le vocabulaire qui sera utilisé dans ce manuscrit.

— Nous appelons synthèse **classique**, ou synthèse **non-incrémentale** le paradigme classique de synthèse "phrase-à-phrase". En synthèse non-incrémentale, l'ensemble des mots constituant la phrase est connu lors de l'analyse par le module de TAL et le module de synthèse. A l'inverse, en synthèse incrémentale, la synthèse d'un mot ne peut s'appuyer que sur les mots déjà saisis, et les mots qu'il reste à saisir sont considérés comme inconnus.

— Nous appelons **mot courant** (respectivement, syllabe, phonème) le mot qui est en cours de traitement par le module de TAL et le module de synthèse.

— En synthèse TTS, l'analyse linguistique et le module de synthèse exploitent traditionnellement des informations contextuelles (ex. nature du phonème suivant, position du mot courant dans la phrase, etc.) Ces informations contextuelles sont encodées dans un vecteur dit de **descripteurs contextuels** (variables continues ou discrètes).

— Dans le cadre de la synthèse incrémentale, on prend le soin de distinguer le **contexte gauche** du **contexte droit**. Le **contexte gauche** comprend tous les descripteurs contextuels qui se rapportent aux mots qui précèdent le mot courant. Ainsi, pour une synthèse phrase-à-phrase, les mots entre le début de la phrase et le mot courant font partie du contexte gauche.

Inversement, les mots entre le mot courant et la fin de la phrase font partie du **contexte droit**. En synthèse incrémentale, la taille du contexte droit est a priori nulle ou très limitée, tandis que le contexte gauche est complètement connu.

La figure 1.8 illustre ce que nous appelons dans le cadre de cette thèse "mot courant", "contexte gauche" et "contexte droit".

1.5 Synthèse vocale incrémentale, réactive, performative

La synthèse vocale incrémentale est une discipline "jeune" qui, à ce jour, a donné lieu à un nombre limité de travaux. Nous regroupons ces derniers en fonction de l'application visée, à savoir la conception de systèmes de dialogue réactifs où la synthèse vocale incrémentale est mise en œuvre en bout de chaîne, la synthèse vocale performative/temps-réel/réactive à but artistique, et le traitement automatique du langage naturel, dont l'objectif n'est pas obligatoirement la synthèse vocale mais, par exemple, la compréhension automatique ou la traduction d'un texte qui se dévoile progressivement. Ces différents travaux sont brièvement décrits dans les sous-sections suivantes. Certains de ces travaux feront l'objet d'une description plus détaillée au cours des chapitres suivants.

FIGURE 1.8 – Illustration de mot courant, contexte gauche et contexte droit. La phrase en cours de saisie est “Cet été, les enfants vont en vacance à la mer” dont les 5 premiers mots (“Cet été, les enfants vont”) ont été saisis. Le cadre vert correspond au **mot courant**. Le cadre rouge représente ce que nous appelons **contexte gauche** : “Cet été les enfants vont”. Il correspond à tout ce qui précède le mot en courant ainsi que le mot courant lui même. Enfin, le cadre bleu représente ce que nous appelons **contexte droit**. Il correspond à tout ce qui suit le mot courant (et qui est donc inconnu dans le cadre de la synthèse incrémentale).

1.5.1 Système de dialogue réactif

L'objectif est ici de concevoir des systèmes de dialogue dont la synthèse vocale est adaptée "en ligne" en fonction de stimuli extérieurs tels que l'environnement, la commande vocale de l'utilisateur ou son comportement (Edlund 2008, Buschmeier et Kopp 2011). Dans (Edlund 2008), la synthèse vocale peut être interrompue par un facteur extérieur (exemple : détection d'un bruit masquant), puis reprise quelques mots avant pour faciliter la compréhension. (Buschmeier et Kopp 2011) propose d'interrompre la diffusion de la parole de synthèse lorsque l'utilisateur ne regarde plus ou s'éloigne de l'interface du système de dialogue (et de l'inviter à revenir).

Cependant, dans ces travaux, l'ensemble du texte à vocaliser est connu au moment de la synthèse. Rappelons ici que dans cette thèse, nous adoptons un paradigme très différent, puisque la synthèse vocale *accompagne* la saisie du texte.

A notre connaissance, la synthèse incrémentale d'une phrase "incomplète" est abordée pour la première fois par (Baumann et Schlangen 2012b) qui proposent, dans le cadre du projet "InPro TK" ("*Incremental Spoken Dialogue Processing Toolkit*", Baumann et Schlangen 2012c), un système complet de dialogue "incrémental". Ce dernier est, par exemple, capable de vocaliser le début d'un message dont la fin ne sera déterminée qu'après l'ajout d'informations supplémentaires. Par exemple, la synthèse de la phrase "la voiture va tourner à droite/gauche" peut débuter avant de connaître la destination finale (i.e. droite ou gauche). La technique de synthèse permettant la prise en compte de ce type d'incertitude dans le module de synthèse sonore (par HMM), décrite dans (Baumann 2014), sera détaillée au Chapitre 3.

1.5.2 Synthèse vocale performative/réactive/temps-réel

L'objectif est ici de concevoir un système de synthèse vocale dont un ou plusieurs paramètres peuvent être contrôlés en temps-réel par l'utilisateur. Dans ce but, l'équipe du Professeur Dutoit (laboratoire TCTS, université de Mons en Belgique / institut Numédiart) a développé le système pHTS/MAGE (Dutoit et al. 2011, Astrinaki 2014). Il s'agit d'une implémentation temps-réel du module de synthèse sonore par HMM "HTS" (HTS 2000). A l'aide de ce système, (Astrinaki 2014) a notamment évalué la perte de qualité d'une synthèse TTS en l'absence totale de contexte droit. Ces travaux sont un point de départ pour cette thèse (les expériences de Baumann et al. et Astrinaki et al., menés principalement en langue Anglaise sont notamment répliqués ici pour la langue française). Dans (Astrinaki 2014), le système pHTS/MAGE permet à l'utilisateur de contrôler en temps-réel la vitesse de synthèse ou le degré d'articulation (hyper/hypoarticulation). On citera également les travaux de (D'Alessandro et Dutoit 2007) sur *Handsketch*, un instrument de musique dont le support est une tablette graphique permettant au musicien d'ajuster en temps-réel et de façon continue la qualité vocale, le pitch ou encore l'intensité. Enfin, l'équipe de C. d'Alessandro (LIMSI, Paris, France) développe également un instrument vocal performatif nommé *Cantor Digitalis*. De nombreux travaux sont menés par cette équipe pour évaluer différentes interfaces de contrôle, telles qu'un joystick, une tablette, etc. (Le Beux 2009 ; Feugère 2013 ;

Perrotin 2015).

1.5.3 Traitement incrémental du langage naturel

Enfin, le traitement "incrémental" du langage naturel a également fait l'objet de plusieurs contributions (en dehors de la synthèse vocale). Ces dernières portent d'une part sur l'analyse morpho-syntaxique incrémentale (en anglais *Part-of-speech tagging*), qui cherche à déterminer la fonction grammaticale de chaque mot (Beuck, Köhn et Menzel 2011), et d'autre part sur l'analyse syntaxique (en anglais *syntactic parsing* ou *prosodic phrasing*), qui vise à établir une description symbolique de la structure syntaxique du texte. Comme détaillé dans (Cadic 2011), l'objectif est de constituer des "unités prosodiques" pouvant prendre la forme d'un groupe accentuel, c'est-à-dire un regroupement de mots par unité de sens, et d'un groupe intonatif, qui est constitué de groupes accentuels appartenant à une même phrase prosodique (Rossi et al. 1981), délimités par exemple par des pauses. Plusieurs techniques ont été proposées pour effectuer l'analyse structurelle de façon incrémentale parmi lesquelles (Mori, Matsubara et Inagaki 2001 ; Rosé, Roque et Bhembe 2002). Dans ce travail de thèse, nous nous sommes néanmoins focalisé sur un seul aspect du TAL, à savoir l'analyse morpho-syntaxique incrémentale. L'analyse structurelle (incrémentale) ne sera pas abordée ici. De plus, comme nous le détaillerons au chapitre 2, le synthétiseur TTS incrémental en langue française basé sur l'approche paramétrique par HMM que nous avons développé, n'utilise pas une description explicite des groupes intonatifs pour la génération du contenu de la parole de synthèse.

1.6 Contributions de ce travail et organisation du document

Les contributions de cette thèse, qui structurent l'organisation de ce document, sont :

— **Un algorithme d'analyse morpho-syntaxique dit à "latence adaptative" pour la synthèse incrémentale**. Cet algorithme exploite une version légèrement modifiée d'un analyseur morpho-syntaxique basé sur l'approche *n-gram*. Une série d'arbres de décision estime la stabilité d'une classe lexicale en fonction du contexte gauche déjà analysé. Si la classe lexicale est jugée instable (i.e est susceptible d'être modifiée lorsqu'un mot suivant sera entré par l'utilisateur), alors l'algorithme décide de retarder la synthèse sonore pour éviter une erreur de phonétisation et/ou une prosodie incorrecte. Cet algorithme, ainsi que son évaluation objective, font l'objet du Chapitre 2.

— **Une méthode de création d'une voix de synthèse de type HMM adaptée à la synthèse incrémentale**. Dans le cadre de la synthèse par Modèles de Markov Cachés (*Hidden Markov Model* en anglais, *HMM* dans ce document), la parole est synthétisée en concaténant des modèles statistiques de phonèmes en contexte. Les descripteurs renseignant sur le contexte droit d'un mot courant utilisés en synthèse non-incrémentale, sont inconnus en synthèse incrémentale. La méthode proposée vise à entraîner des modèles en prenant en compte cette incertitude sur la valeur des descripteurs associés au contexte droit. Cette prise en compte s'effectue notamment au moment de l'étape de

regroupement des états HMM pour l'estimation robuste des paramètres (*state-tying*), et consiste à regrouper des modèles (et donc des contextes linguistiques) partageant la même incertitude sur la valeur d'un ou plusieurs descripteurs contextuels. Cette méthode d'apprentissage des modèles, ainsi que ses évaluations objectives et perceptive, font l'objet du Chapitre 3.

— **Un prototype complet d'un système TTS incrémental pour la langue française, basé sur ces deux méthodes, ainsi que son évaluation perceptive**. Ce prototype est décrit au Chapitre 4. Ce logiciel a été développé dans le cadre du projet *SpeakRightNow*[7] (financé par l'Université Grenoble-Alpes) en collaboration avec des ergothérapeutes. Il vise, à terme, à être utilisé comme outil de suppléance vocale par des personnes en situation de handicap, et a donc été développé pour fonctionner sur tablette ou smartphone, et se base sur une architecture de type "*cloud computing*".

A ce jour, ce travail de thèse a fait l'objet de deux publications dans les actes de la conférence internationale Interspeech (Pouget et al. 2015 ; Pouget et al. 2016). Ces publications figurent à la fin de ce document.

7. site du projet : `http://www.gipsa-lab.fr/projet/SpeakRightNow/`

Chapitre 2

Traitement Automatique de la Langue naturelle pour la synthèse incrémentale de la parole

Sommaire

2.1 Introduction

Les différents traitements réalisés au sein d'une chaîne de synthèse de parole à partir du texte se divisent en deux modules principaux : le Traitement Automatique de la Langue naturelle (que nous appellerons par la suite "module de TAL", dont le présent chapitre fait l'objet) permettant de convertir le texte en entrée en une séquence de phonèmes accompagnés de descripteurs contextuels et le module de synthèse sonore permettant de synthétiser le signal de parole.

Au cours de ce chapitre, nous commençons par rappeler brièvement les différents traitements effectués par un module de TAL (dans le cadre de la synthèse TTS). Nous décrivons ensuite les verrous à lever pour effectuer chacun de ces traitements dans un contexte de synthèse TTS incrémentale. Enfin, nous nous focalisons sur un de ces traitements, à savoir l'analyse morpho-syntaxique (*Part-of-Speech (POS) tagging* en anglais) et nous proposons une méthode pour l'adapter à la synthèse TTS incrémentale.

2.2 Fonctionnement général d'un module de TAL pour la synthèse TTS

Dans cette section, nous rappelons brièvement les différents traitements effectués classiquement par le module de TAL dans un système TTS.

Ces différents traitement sont illustrés à la Figure 2.1.

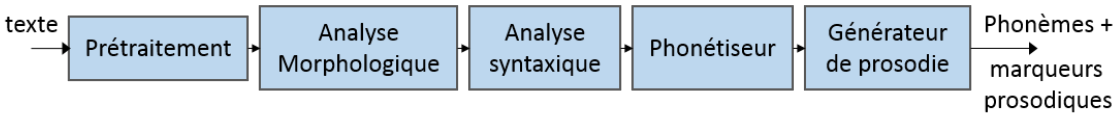


FIGURE 2.1 – Diagramme fonctionnel représentant les différentes étapes du module de Traitement Automatique de la Langue naturelle inspiré de (BOITE et al. 2000).

2.2.1 Prétraitement

Le pré-traitement consiste à segmenter la phrase à synthétiser en "unités lexicales". Il s'agit de convertir en toutes lettres les abréviations (Mr.), les acronymes (SMIG), les chiffres et nombres (42), les heures (2h20), les symboles ($). Par exemple, le résultat du prétraitement de la phrase "M. Martin a RDV à 8h" donnera après segmentation "Monsieur Martin a rendez-vous à huit heures.". (INDURKHYA, DAMERAU et PALMER 2010) traitent les différentes ambiguïtés qui peuvent émerger lors du prétraitement parmi lesquels : l'identification de traits d'union dans des mots tels que "c'est-à-dire" par rapport aux traits d'unions permettant de découper un mot en fin de ligne ; la virgule en temps que ponctuation par rapport à la délimitation entre partie entière et partie numérique d'un nombre ; l'identification de locutions adverbiales telles

que "de temps en temps" qui doivent être considérées comme une seule unité. Le prétraitement peut se faire grâce à divers outils tels que Lex (Lesk et Schmidt 1975), Flex (Nicol 1993).

2.2.2 Analyse morphologique

À l'issue du prétraitement, la phrase est une séquence de mots transcrits en toutes lettres. L'analyse morphologique consiste à déterminer les classes lexicales possibles de chaque mot de la phrase à synthétiser (nom, verbe, préposition, adverbe, pronom, etc.). À ce stade de l'analyse, plusieurs classes lexicales peuvent être envisagées pour un même mot. Aussi, deux classes seront à ce stade associées au mot "président" : i) un verbe ("présider" conjugué à la troisième personne du pluriel du présent de l'indicatif ou du subjonctif) et ii) un nom, (personne qui assure la présidence). Notons que la prononciation de "président" dépend de sa classe lexicale.

On distingue deux catégories de mots lors de l'analyse morphologique : les mots grammaticaux (déterminants, conjonctions, pronoms, prépositions) et les mots lexicaux. Les mots grammaticaux servent à définir la structure de la phrase. Ils sont en nombre fini (moins de 1000) et peuvent donc être indexés dans un dictionnaire. Les mots lexicaux, à l'inverse, sont a priori, en nombre infini, l'évolution d'une langue impliquant justement la création de mots lexicaux (le mot "incrémentalité" étant un exemple de néologisme utilisé dans ce document).

La création de mots lexicaux obéit à des règles d'appartenance d'unités signifiantes minimales appelées morphèmes ou lexèmes : les bases (ou formes canoniques), les suffixes, les préfixes. L'analyse *morphologique* permet de glaner les attributs propres à chaque morphème et ainsi de calculer les attributs potentiels des mots. Un *morphème* est une unité de sens minimale renseignant sur une propriété d'un mot. Ces propriétés peuvent par exemple être grammaticales ("pluriel"), lexicales ("adverbe") ou sémantiques ("écrire"). On se sert des morphèmes lors de l'analyse morphologique pour décomposer un mot en unités minimales. Par exemple (inspiré de Yvon 2010), l'analyse morphologique du mot "président" peut se faire de deux façons différentes :

— présider - verbe - 3^{eme} personne - pluriel - présent - indicatif ou subjonctif

— présider (action de) - nom - masculin - singulier

On décrit donc un mot lexical par un lexème et par les morphèmes grammaticaux qui le composent et qui peuvent agir sur le lexème selon trois principales opérations morphologiques : la morphologie flexionnelle qui concerne par exemple l'accord des noms ou la conjugaison des verbes, la morphologie dérivationnelle qui, à l'aide de suffixes ou d'affixes permet de changer la nature (la terminaison -el à un nom tel que *constitution* permet de le transformer en adjectif : *constitutionnel*) ou le sens (l'ajout du préfixe *anti-* permet d'en inverser le sens : *anticonstitutionnel*) et la morphologie compositionnelle qui permet de combiner des morphèmes de sens entre eux pour parvenir à des mots tels que *taille-crayon* ; *auto-école* ou *pomme de terre*.

L'analyse morphologique permet d'exploiter toutes les connaissances lexicales dont on dispose pour caractériser les mots d'une phrase, elle ne fournit cependant pas d'informations

sur les relations entre ces mots.

2.2.3 Analyse syntaxique

L'analyse syntaxique (ou analyse morpho-syntaxique) vise à déterminer de façon univoque la classe lexicale de chaque mot de la phrase à synthétiser (rappelons qu'un ensemble de classes possibles a été déterminé lors de l'analyse morphologique). Cette analyse va notamment s'appuyer sur l'analyse du contexte de chaque mot, qui va aider à lever une ambiguïté éventuelle sur sa classe lexicale.

2.2.3.1 État-de-l'art

La problématique de l'analyse syntaxique automatique a vu au cours du temps son traitement évoluer. Si l'attribution de classes lexicales pour les mots d'une phrase s'est dans un premier temps faite par règles (Greene et Rubin 1971), les approches probabilistes (Jelinek et Chelba 1999), plus facilement adaptables d'une langue à une autre, les ont remplacées.

Aussi, nous restreignons cet état de l'art à ces dernières.

Une approche classique utilisée pour l'analyse syntaxique repose sur la modélisation *n-gram*. Il s'agit d'exploiter la probabilité d'observer une classe lexicale sachant les classes lexicales des n mots précédents (la séquence de classes lexicales associée à une séquence de mots se modélise alors par un processus Markovien) [1]. C'est notamment l'approche utilisée par *TnT* (Brants 2000) qui atteignait en 2000 une précision de 96,7% sur l'analyse du corpus de l'université de Pennsylvanie : *Penn Treebank* (Marcus, Marcinkiewicz et Santorini 1993). Ses travaux font encore aujourd'hui valeur de référence. Il s'agit par ailleurs de la méthode utilisée par les synthétiseurs TTS *festival* (Taylor, Black et Caley 1998) et *maryTTS* (Schröder et Trouvain 2003) pour l'analyse syntaxique. C'est également le cas du synthétiseur COMPOST (Bailly et Alissali 1992) utilisé dans le cadre de ce travail. Aussi, cette approche par modélisation n-gram sera décrite plus en détails à la Section 2.4.1.

(Giménez et Marquez 2004) ont proposé une approche basée sur les Machines à Vecteur Support (*Support Vector Machines*, ou *SVM* en anglais) : SVMTool réalisant l'analyse du corpus *WSJ* avec une précision de 97,16%, contre 96,46% pour *TnT* sur le même corpus.

Plus récemment, (Sokolovska et al. 2010) et (Lavergne, Cappé et Yvon 2010) ont proposé une approche basée sur les Champs Markoviens Aléatoires (*Conditionnal Random Fields*, ou CRF). Cette méthode, également utilisée par (Constant et al. 2011) et (Sun 2014) permet d'analyser le corpus *Penn Treebank* (Marcus, Marcinkiewicz et Santorini 1993) avec une précision de 97,36%.

Enfin, (Collobert et al. 2011) dans un premier temps puis (Santos et Zadrozny 2014) ensuite ont proposé d'utiliser une approche de type *deep learning* basée sur des réseaux à convolution pour résoudre simultanément les différents traitements d'une chaîne de TAL (ana-

1. (Toutanova et al. 2003) proposent une architecture exploitant également explicitement les relations entre un mot et les suivants

lyse syntaxique, *parsing*, etc.). Leur approche consiste à entraîner les réseaux à convolution sur des corpus constitués de données annotées (corpus Reuters RCV1, (Lewis et al. 2004) de 231 millions de mots) et de données non-annotées (631 millions de mots, issus de *Wikipedia*). Le réseau extrait automatiquement une série d'abstractions qui renseignent sur la structure morphologique, syntaxique, voire sémantique d'une chaîne de caractère. Pour l'analyse morpho-syntaxique, cette approche permet d'atteindre une précision de 97,32% pour l'anglais.

2.2.4 Traitements sémantiques et pragmatiques

À l'issue de l'analyse syntaxique, une classe lexicale est attribuée à chaque mot de la phrase. À ce stade de l'analyse, les contenus syntaxiques et lexicaux sont connus mais le sens que véhicule l'énoncé reste inconnu. Le traitement sémantique permet de s'intéresser au sens d'un énoncé. Une phrase telle que :

L'extincteur raconte le chat.

quoique correcte sur les plans lexicaux (tous les mots sont dans un dictionnaire) et syntaxiques (Déterminant (l') - Nom (extincteur) - Verbe (raconte) - Déterminant (le) - Nom (chat)) et même grammaticaux (sujet - verbe - complément d'objet), n'a pas de sens dans la plupart des contextes.

Le traitement sémantique a pour objectif de formaliser les relations entre les différents objets d'un énoncé. Ce traitement permet donc de désambiguïser des mots polysémiques. Ainsi, dans des phrases telles que

Il commande un whisky.

et

Il commande un navire.

Identifier *whisky* comme étant une boisson et *navire* comme étant un véhicule permet de distinguer les différents sens de commander : *demander* (dans un restaurant) et *conduire/diriger* un véhicule.

Bien que dans le cadre de la synthèse de parole, la fonction première du TAL est de fournir au module de génération de la forme d'onde une séquence de phonèmes et de descripteurs contextuels et non pas d'interpréter un énoncé (comme cela pourrait être le cas pour de la traduction automatique ou de la reconnaissance), l'analyse sémantique peut cependant s'avérer nécessaire pour déterminer la séquence de phonèmes à synthétiser lorsque des homographes hétérophones appartiennent à la même classe lexicale. Par exemple, Considérons les énoncés suivants.

Ses fils sont branchés. Ils portent des vêtements à la mode

et

Ses fils sont branchés. Le courant va pouvoir circuler.

Dans les deux cas, le mot *fils* est un nom pluriel, mais la prononciation sera différente s'il s'agit d'un câble /fil/ ou des enfants de quelqu'un /fis/. La désambiguïsation entre les

deux prononciations nécessite une analyse sémantique de l'énoncé dans lequel cette phrase est insérée.

Il reste enfin un niveau d'analyse supérieur à l'analyse sémantique : l'analyse pragmatique. L'analyse pragmatique permet d'intégrer à une phrase les connaissances sur les locuteurs et l'environnement qui ne sont pas explicitement spécifiées dans l'énoncé. L'exemple suivant permet d'en comprendre le principe :

Pierre : Tu viens au bal ce soir ?
Marie : J'ai entendu dire que Mathieu y sera !

Dans cette phrase, seules les connaissances de Pierre sur la relation entre Mathieu et Marie permettront de savoir si la réponse de Marie veut implicitement dire *oui* ou *non*. L'analyse pragmatique permet par exemple la mise en place de synthèse d'interjections ou de backchanneling dans des systèmes de dialogue (McTear 2004).

2.2.5 Phonétisation

L'étape de phonétisation (également appelée conversion lettre-son ou graphème-phonème) consiste à prédire la prononciation d'un mot à partir de son orthographe en convertissant une séquence de lettres en une séquence de phonèmes.

Comme le constatent (Toma et al. 2013), les méthodes de phonétisation automatiques peuvent se décomposer en trois classes distinctes : les approches par dictionnaire, les approches par règles et les approches par apprentissage automatique.

— Les approches par dictionnaire consistent à disposer d'un large lexique contenant un maximum de mots d'une langue accompagnés de leur prononciation. Cette approche consiste soit à stocker l'ensemble des formes canoniques et leurs formes fléchies avec leur prononciation, soit à stocker des morphèmes ainsi que des règles permettant d'inférer la prononciation de leurs combinaisons. Cette approche a été utilisée par (Allen, Hunnicut et Klatt 1987) et possédait un dictionnaire disposant de 12 000 morphèmes. Le système de phonétisation automatique d'AT&T reposait également sur une approche par un dictionnaire (Levinson, Olive et Tschirgi 1993) contenant plus de 43 000 morphèmes.
— Les approches par règles consistent à appliquer un ensemble de règles phonétiques établies par des experts phonéticiens pour déterminer la prononciation d'un mot. Ces approches présentent l'inconvénient d'être spécifiques à chaque langage (et même à chaque dialecte) et doivent être adaptées lorsque des mots étrangers tels que *revolver* ou *sushi* sont introduits dans une langue. Des méthodes reposant sur des approches par règles peuvent être trouvées dans (Toma et Munteanu 2009) ou (Braga, Coelho et Resende 2006).
— Les approches par apprentissage automatique permettent d'apprendre de façon automatique les relations graphèmes-phonèmes grâce à l'analyse de corpus annotés. Elle reposent sur divers algorithmes (dont certains ont été évoqués en partie 2.2.3 pour

l'analyse morpho-syntaxique). Le problème de la génération de phonèmes à partir de lettres présente deux principales difficultés : l'alignement et la conversion.
Le premier problème rencontré dans le processus de phonétisation automatique est d'aligner une séquence de phonèmes et une séquence de lettres. Ainsi, l'alignement de la séquence /ɛgzɑ̃pl/ avec les lettres *exemple* peut être problématique dans la mesure où une lettre peut correspondre à un phone (**e**xemple → ɛ), une lettre peut correspondre à plusieurs phones (e**x**emple → gz) et où une lettre peut ne pas correspondre à un son : (exampl**e** → -). (Jiampojamarn et Kondrak 2010) proposent une revue de méthodes d'alignement lettres-phonèmes.
Le second problème rencontré dans la phonétisation automatique concerne la transcription lettres vers sons. (Pagel, Lenzo et Black 1998) proposent d'utiliser pour cette étape des arbres de décision prenant en entrée une fenêtre glissante de lettres et associant à chaque lettre, soit un phonème unique, soit un phonème double (e**x**emple → gz), soit un phonème nul. Ils parviennent à obtenir une précision de 61,4% de mots correctement phonétisés, soit une précision de 87,9% sur les phonèmes, sur le corpus de l'université de Carnegie-Melon (*CMU*, Kominek et Black 2004). (Che, Tao et Pan 2012) proposent de résoudre ce problème à l'aide de modèles de Markov cachés couplés permettant de trouver de manière simultanée, à partir d'une chaîne de caractères, le regroupement optimal des lettres et la transcription phonétique de ces groupes. Ils parviennent ainsi à un taux de bonnes réponses (par mot) de respectivement 74,6% et 94,2% sur les corpus *CMU* et le corpus "Oxford Advanced Learner's Dictionnary of Current English" (OALD, Mitton 1992). (Wang et King 2011) puis (Hahn et al. 2013) proposent l'utilisation de Conditional Random Fields (Champs de Markov Aléatoires, basés sur des modèles graphiques orientés). Ils parviennent, grâce à l'utilisation des CRF à un taux de bonnes réponses de 84,6% sur le corpus QUAERO (Sundermeyer et al. 2011). Enfin le développement des réseaux de neurones profonds a également permis la mise au point d'algorithmes reposant sur les *Long Short-Term Memory* (*LSTM*) *Recurrent Neural Network*, proposé par (Rao et al. 2015). Cette technique propose de se passer d'alignement et permet de faire de la transcription lettre-son avec une précision de 78,7% sur le corpus CMU.

2.3 Verrou technologique à lever pour un TAL incrémental

Dans cette section, on entend par "traitement *incrémental* de la langue naturelle", la réalisation de la chaîne de TAL mentionnée précédemment (pré-traitement, analyse morphologique, syntaxique, phonétisation, etc.) à partir d'un texte qui se dévoile progressivement. Dans le cadre de la synthèse TTS incrémentale, il s'agit de traiter une phrase au fur et à mesure de sa saisie par l'utilisateur. Un des principaux challenges que pose le paradigme d'incrémentalité est l'absence d'information sur le "contexte droit" d'un mot saisi. Rappelons que la plupart des étapes d'une chaîne de TAL exploitent les contextes gauche et droit pour lever une ambiguïté sur la classe lexicale et la phonétisation d'un mot.

Pré-traitement Si la segmentation de la chaîne de caractères en entrée du module de prétraitement ne semble pas problématique dans le cadre de la saisie incrémentale, l'étape du module de prétraitement consistant à identifier les locutions adverbiales devient problématique en l'absence de contexte droit. En effet, la reconnaissance d'une locution adverbiale telle que "de temps en temps" (et notamment la réalisation de la liaison médiane) nécessite l'attente de la saisie de l'ensemble des termes qui la composent.

Analyse morpho-syntaxique L'analyse syntaxique s'appuie sur une optimisation sur l'ensemble de la phrase. Dans le contexte de la synthèse incrémentale, l'absence de connaissances sur le contexte droit est donc critique pour cette étape. À notre connaissance, la littérature sur l'analyse morpho-syntaxique incrémentale est assez restreinte. (Beuck, Köhn et Menzel 2011) apportent néanmoins des premières pistes de réflexion sur l'analyse morpho-syntaxique. Les auteurs discutent différentes stratégies :

— Effectuer l'analyse en s'appuyant uniquement sur le contexte gauche. Le système est donc à latence nulle (aucun temps entre la saisie du mot et la détermination de sa classe lexicale).
— Reporter l'analyse d'un mot le temps que l'utilisateur saisisse quelques mots supplémentaires pour lever une ambiguïté sur sa classe lexicale. Cette approche introduit une latence que les auteurs décrivent a priori comme fixe.
— Reconsidérer **a posteriori** la classe lexicale attribuée à un mot après avoir déjà attribué une classe lexicale aux mots qui le suivent. Cette approche est bien adaptée à des contextes applicatifs comme la traduction automatique pour laquelle il est envisageable de mettre à jour un texte déjà traduit. En revanche, il apparaît plus difficile de la mettre en œuvre dans le cadre de la synthèse incrémentale car un mot déjà vocalisé ne peut être modifié a posteriori.
— Proposer plusieurs alternatives pour la classe lexicale d'un mot, en associant éventuellement à chaque alternative une certaine probabilité. Ces probabilités sont ensuite propagées aux autres modules de la chaîne de TAL (ex. : le phonétiseur).

Phonétisation En supposant correcte l'analyse morpho-syntaxique, l'étape de phonétisation requiert souvent la connaissance du contexte droit, en français par exemple, pour la prononciation des consonnes latentes comme les liaisons. Dans le cadre de ce travail, nous effectuerons la synthèse de celles-ci avec le mot suivant lorsqu'elles sont réalisées.

Dans ce travail, nous nous penchons principalement sur la réalisation de l'analyse syntaxique dans le cadre de la synthèse incrémentale. Pour ce faire, nous nous appuyons sur les concepts introduits par Beuck, Köhn et Menzel 2011. Nous proposons de déterminer automatiquement la taille du contexte droit nécessaire pour garantir l'analyse correcte d'un mot (ex. : sa classe lexicale ou sa phonétisation). Dans notre approche, la taille du contexte droit peut être variable. Le synthétiseur résultant est donc à latence variable mais bornée. Cette approche vise à trouver un compromis entre réactivité du synthétiseur et précision. Notre méthode a été mise en œuvre et évaluée dans le cadre de l'analyse morpho-syntaxique incrémentale. Elle est décrite à la Section 2.4.2.

2.4 Méthode proposée pour l'analyse morpho-syntaxique incrémentale

Dans cette section, nous décrivons la méthode que nous proposons pour effectuer l'analyse morpho-syntaxique dans un contexte incrémental. Notre méthode s'appuyant sur une analyse morpho-syntaxique basée sur la modélisation n-gram (mentionnée brièvement dans la Section 2.2.3), nous commençons par en rappeler le principe général, et discutons de son utilisation dans un contexte incrémental.

2.4.1 Rappel sur le fonctionnement d'un analyseur morpho-syntaxique basé sur l'approche n-gram

L'analyse morpho-syntaxique attribue à chaque mot d'une séquence de N mots $\mathbf{M} = [m_1,\ m_2, ..., m_N]$, une séquence de classes lexicales $\hat{\mathbf{C}} = [\hat{c}_1,\ \hat{c}_2, ..., \hat{c}_N]$ où $\hat{c}_{i\in[1,N]}$ prend une valeur dans un ensemble de K classes lexicales possibles [2].

Le problème de l'analyse syntaxique peut donc se formuler de façon probabiliste :

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} P(\mathbf{C}|\mathbf{M}) \tag{2.1}$$

qui se reformule grâce au théorème de Bayes par :

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} \frac{P(\mathbf{M}|\mathbf{C})P(\mathbf{C})}{P(\mathbf{M})} \tag{2.2}$$

Le dénominateur ne dépendant pas de $\mathbf{C}$, il peut être ignoré pour la recherche de $\hat{\mathbf{C}}$, on obtient donc :

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} P(\mathbf{M}|\mathbf{C})P(\mathbf{C}) \tag{2.3}$$

La modélisation n-gram repose sur les deux hypothèses suivantes :

— H1. La probabilité d'observer le mot m_i de $\mathbf{M}$ sachant la séquence complète de classes lexicales $\mathbf{C}$ ne dépend que de la classe lexicale c_i associée à ce mot et non des classes associées aux autres mots de la séquence. Cette hypothèse permet de simplifier le premier terme de l'Équation 2.3 $P(\mathbf{M}|\mathbf{C})$.

— H2. La probabilité d'observer la classe lexicale c_i de $\mathbf{C}$ peut être estimée uniquement à partir des n classes précédentes. Cette hypothèse fait référence au second terme de l'Équation 2.3 $P(\mathbf{C})$.

2. Le nombre K de classes lexicales peut varier selon l'application que l'on envisage pour le texte à analyser. Il peut aller d'environ 10, dans les manuels scolaires de français, à une trentaine pour le traitement automatique du français. En anglais, K varie d'une dizaine (nom, pronom, adjectif, verbe, adverbe, préposition, conjonction, interjection, article) à 36 classes pour l'annotation du corpus de l'université de Pennsylvanie, PennTreeBank (Marcus, Marcinkiewicz et Santorini 1993).

En prenant en compte ces deux hypothèses, l'Équation 2.3 devient :

$$\hat{\mathbf{C}} \approx \arg\max_{\mathbf{C}} \prod_{i=1}^{N} P(m_i|c_i)P(c_i|c_{i-1}, ...c_{i-n}) \tag{2.4}$$

Dans le cas d'une modélisation 3-gram (sur laquelle est basé l'analyseur syntaxique utilisé dans nos expériences), cette équation devient :

$$\hat{\mathbf{C}} \approx \arg\max_{\mathbf{C}} \prod_{i=1}^{N} P(m_i|c_i)P(c_i|c_{i-1}, c_{i-2}) \tag{2.5}$$

Les probabilités $P(m_i|c_i)$ et $P(c_i|c_{i-1}, c_{i-2})$ sont souvent estimées sur un large corpus de textes annotés par comptage d'occurrences. Par exemple, on comptera le nombre de fois où le mot "livre" est associé à la classe "Nom" et à la classe "Verbe" pour obtenir la quantité P(livre|Nom) et P(livre|Verbe). De façon similaire, on estimera les probabilités de transition par comptage d'occurrences de chaque séquence possible de trois classes lexicales (exemple : P(Nom|Verbe, Adjectif), P(Verbe|Adjectif, Nom)).

Le problème majeur de cette approche fréquentiste est que le corpus d'apprentissage ne couvre généralement pas toutes les successions de classes lexicales possibles. La probabilité associée aux séquences non observées est alors nulle. Ces séquences ne sont donc jamais considérées comme séquences potentielles lors de l'analyse syntaxique.

Ce problème dit de dispersion des données (*data sparsity*) se traite grâce à des techniques dites *de lissage*. Le but du lissage est de rendre la distribution des séquences observées plus uniforme en attribuant une probabilité non nulle aux séquences non observées et en ajustant à la baisse les probabilités trop fortes des séquences trop fréquentes. Par soucis de concision, nous ne détaillerons pas les différents algorithmes de lissage possibles. Une revue de la littérature sur ce sujet est disponible dans (Chen et Goodman 1999).

Le problème de l'analyse syntaxique décrit par l'Équation 2.4 peut être modélisé par un modèle de Markov défini par :

- Un ensemble de Q états avec classiquement $Q = K + 2$, correspondant aux K classes lexicales possibles auxquelles on ajoute des états mentionnant le début et la fin de la phrase (notons dès à présent que ce dernier n'est a priori pas accessible en synthèse incrémentale (sauf en fin de phrase) et qu'une adaptation spécifique sera donc nécessaire). Par soucis de simplification des notations, on notera cet ensemble $\{q_0, q_1, ..., q_k, ..., q_K, q_{K+1}\}$, c'est-à-dire en utilisant la même notation q_k pour le k^{eme} état, et la classe lexicale qu'il représente. q_0 et q_{K+1} sont les marqueurs de début et de fin de phrase.
- Un ensemble de probabilités de transition d'une classe à l'autre $a_{ijk} = P(q_i|q_j, q_k)$ (dans le cas des 3-gram). Il s'agit ici de probabilités *a priori* car indépendantes de la séquence de mots observée.
- Une probabilité d'émission associée à l'état q_k, $b_k(m_i) = P(m_i|c_i = q_k)$ qui est la probabilité d'observer le mot m_i sachant la classe lexicale associée c_i prend la valeur q_k. Par souci de concision, on notera plus simplement $b_k(m_i) = P(m_i|q_k)$.

En s'appuyant sur ce formalisme, la résolution du problème d'analyse syntaxique, c'est-à-dire la résolution de l'Équation 2.4, peut se faire classiquement par programmation dynamique à l'aide de l'algorithme de Viterbi (FORNEY 1973). Ce dernier est illustré en pseudo-code, dans le cas plus simple d'une modélisation 2-gram (Algorithme 1).

Data:
- Séquence de N mots à analyser $\mathbf{M} = [m_1, ..., m_N]$
- Ensemble des K+2 états associés aux classes lexicales possibles auxquelles on rajoute les marqueurs de début et fin de phrase q_0 et q_{K+1} soit $\mathbf{Q} = \{q_0, ..., q_K + 1\}$.
- Probabilités de transition (ici dans le cas d'un 2-gram) a_{ij}=$P(q_i|q_j)$
- Probabilités d'émission (vraisemblance) $b_k(m_i) = P(m_i|c_i = q_k) = P(m_i|q_k)$
- Matrice des distances D de taille $(K + 2) \times N$
- Matrice des pointeurs R de taille $(K + 2) \times N$

for $k \leftarrow 1$ **to** K **do**
  $D(k, 1) = a_{0k} b_k(m_1)$
  $R(k, 1) = q_0$
for $i \leftarrow 2$ **to** N **do**
  for $k \leftarrow 1$ **to** K **do**
    $D(j, i) = \max_k\{D(k, i-1) a_{kj} b_j(m_i)\}$
    $R(j, i) = \text{argmax}_k\{D(k, i-1) a_{kj} b_j(m_i)\}$
—**Procédure de *backtracking***—
$r_N = \text{argmax}_k\{D(k, N)\}$
$\hat{c}_N = q_{r_N}$
for $i \leftarrow N$ **to** 2 **do**
  $r_{i-1} = R(r_i, i)$
  $\hat{c}_{i-1} = q_{r_{i-1}}$
Retourner la séquence de classe lexicale estimée $\hat{\mathbf{C}} = [\hat{c}_1, ..., \hat{c}_N]$

Algorithm 1: Algorithme de Viterbi pour l'analyse syntaxique

L'algorithme de Viterbi fournit donc la séquence $\hat{\mathbf{C}}$ qui maximise la probabilité a posteriori $P(\mathbf{C}|\mathbf{M})$ sachant la séquence de mot entière $\mathbf{M}$. Il se décompose en deux phases. Une première phase qui détermine, pour chaque mot de la séquence, et pour chaque état, la probabilité que cet état appartienne à la séquence d'état la plus probable $\hat{\mathbf{C}}$. Dans la seconde phase dite de rétropropagation (ou *backtracking*), la séquence d'état la plus probable $\hat{\mathbf{C}}$ est déterminée en partant du dernier état, et en considérant récursivement les états prédécesseurs les plus probables.

Pour garantir que cet algorithme fournisse la séquence d'état optimale, la programmation dynamique impose que les états de début et de fin soient connus (*i.e.* $\hat{c}_0 = q_0$ et $\hat{c}_N = q_{K+1}$). En synthèse TTS classique, ces états sont donnés par les marqueurs de début et de fin de phrase (par exemple, par la ponctuation). En synthèse incrémentale, le marqueur de fin de phrase est a priori absent du texte à analyser, car ce dernier se dévoile au fur et à mesure de la saisie (sauf lorsque l'utilisateur termine effectivement sa phrase).

Une première solution simple pour pallier cette absence de marqueur de fin de phrase est de débuter la procédure de *backtracking* à chaque fois qu'un nouveau mot m_i est saisi par l'utilisateur, en ne considérant que les chemins qui partent de l'état le plus probable pour ce mot, c'est-à-dire $\hat{c}_i = q_{r_i}$ avec $r_i = \text{argmax}_k\{D(k,i)\}$. Cependant, cette solution ne garantit pas que la séquence estimée $\hat{\mathbf{C}}$ soit optimale (par rapport aux classes lexicales que l'on obtiendrait en effectuant l'analyse de la phrase à vocaliser, une fois cette dernière complètement saisie). Plusieurs approches ont donc été proposées dans la littérature pour retrouver cette optimalité lors d'un décodage de Viterbi "en ligne" (notamment dans le domaine des télécommunications, ou de la reconnaissance automatique de la parole). On citera par exemple (Seward 2003) qui propose d'effectuer le *backtracking* à intervalle régulier sur une fenêtre glissante de D observations, et de ne considérer que les premiers états décodés. Cette approche est intéressante mais introduit dans notre cas une latence fixe (de D mots) entre la saisie d'un mot et sa synthèse. Par ailleurs, elle ne garantit toujours pas l'optimalité. Une autre approche, appelé *short-term Viterbi* est décrite dans (Bloit et Rodet 2008). Elle considère en parallèle plusieurs séquences d'états possibles et débute le *backtracking* dès qu'un point dit de fusion entre plusieurs séquences alternatives est trouvé. Sous certaines conditions, cette approche garantit l'optimalité de la séquence d'état décodée.

Cet algorithme, proposé initialement pour le décodage acoustico-phonétique, pourrait être envisagé pour l'analyse morpho-syntaxique incrémentale. Cependant, (Bloit et Rodet 2008) rapportent que la latence minimale de cette approche est non nulle. Cela n'est pas un problème en décodage acoustico-phonétique car une nouvelle observation est typiquement rendue disponible toutes les 5 à 10ms (dans le cas d'une analyse par fenêtre glissante de type MFCC). En revanche, dans un contexte de TAL (où une observation représente un mot), cela implique donc qu'un mot ne pourra jamais être synthétisé immédiatement après sa saisie. Dans ce travail de thèse, nous proposons une autre approche basée sur une latence variable mais pouvant être potentiellement nulle. Cette nouvelle approche pour l'analyse morpho-syntaxique incrémentale est décrite à la section suivante.

2.4.2 Méthode proposée : Analyse morpho-syntaxique à latence adaptative

L'idée générale de la méthode proposée est d'estimer, pendant la saisie, si la classe lexicale du mot courant m_i, décodée en considérant la séquence $[m_0, ..., m_{i-1}, m_i]$, est susceptible d'être modifiée (à juste titre) si on considère un contexte droit d'un ou plusieurs mots supplémentaires (c'est-à-dire m_{i+1}, $[m_{i+1}, m_{i+2}]$, $[m_{i+1}, m_{i+2}, m_{i+3}]$, etc.). Si c'est le cas, alors la synthèse du mot courant m_i est retardée le temps d'accumuler le contexte droit nécessaire pour garantir que sa classe lexicale soit correctement estimée.

Pour estimer la stabilité de la classe lexicale d'un mot courant, en fonction de son contexte gauche et d'un contexte droit de taille variable (et potentiellement nulle), nous proposons une approche par apprentissage statistique supervisé, basée sur un ensemble de trois arbres de décision [3]. Un arbre de décision est un outil d'aide à la décision représentant un ensemble

3. Le choix d'utiliser **3** arbres de décisions n'est pas motivé par l'utilisation de 3-gram pour l'estimation des classes lexicales mais par des mesures de performances pour 2, 3 et 4 arbres de décision (voir Section 2.5.3)

de choix sous la forme graphique d'un arbre. Dans notre cas, il s'agit d'un arbre de décision binaire, décidant si la classe $\hat{c}_i$ est jugée stable ou instable étant donné une fenêtre glissante de trois classes lexicales, incluant potentiellement une information sur le contexte droit d'une ou deux classes au maximum. Le premier arbre décide si la classe lexicale $\hat{c}_i$ du mot m_i estimée à partir de $[m_0, ..., m_{i-1}, m_i]$ est stable, sachant les deux dernières classes les plus probables décodées $\hat{c}_{i-2}$ et $\hat{c}_{i-1}$ et aucun contexte droit. Le second évalue cette stabilité en considérant $[\hat{c}_{i-1}\hat{c}_i\hat{c}_{i+1}]$, c'est-à-dire avec un contexte droit d'un mot. Le troisième évalue cette stabilité en considérant $[\hat{c}_i\hat{c}_{i+1}\hat{c}_{i+2}]$, soit deux mots de contexte droit [4].

Ces trois arbres sont utilisés en cascade. Illustrons leur fonctionnement par un exemple, en suivant le parcours du mot "grand" dans la synthèse incrémentale de la séquence "Il est grand", "grand" étant le dernier mot tapé par l'utilisateur. Le premier arbre évalue la stabilité de la classe [Adjectif] sachant le contexte lexical [Pronom – Auxiliaire – Adjectif] associé à "Il est grand", soit un contexte gauche de deux mots. Si elle est jugée stable, la synthèse de "grand" est déclenchée, sinon, elle est retardée. L'utilisateur rentre alors le mot "et". Le second arbre évalue alors la stabilité de la classe [Adjectif] dans le contexte [Verbe – Adjectif – Conjonction] associée à "est grand et". Si elle est jugée stable, alors le mot "grand", mis en attente, et synthétisé. Sinon, il reste en attente. L'utilisateur rentre un nouveau mot "sympathique". Le troisième arbre évalue alors la stabilité de la classe [Adjectif] en considèrent le contexte [Adjectif – Conjonction – Adjectif] associé à la séquence "grand et sympathique". Si elle est jugée stable, alors le mot "grand" est synthétisé. Dans le cas contraire, il sera systématiquement synthétisé à l'ajout du mot suivant, donc avec une latence entre la saisie et la synthèse au maximum de trois mots.

Les procédures utilisées pour l'estimation en ligne des classes lexicales et pour l'apprentissage des arbres de décision seront respectivement détaillées aux Sections 2.4.3.1 et 2.4.3.2.

La Figure 2.2 schématise le fonctionnement général de la méthode proposée.

La latence obtenue par cette méthode est donc variable [5] : elle dépend des décisions prises par les différents arbres de décision. Notons qu'elle peut être nulle lorsque le premier arbre considère que la classe lexicale ne changera pas même si on considère un ou plusieurs mots de contexte droit.

2.4.3 Implémentation

2.4.3.1 Analyse morpho-syntaxique en ligne

La méthode proposée nécessite une analyse morpho-syntaxique "en ligne", c'est-à-dire au fur et à mesure de la saisie du texte par l'utilisateur. Dans notre implémentation, cette analyse est effectuée à l'aide d'une modélisation de type n-gram tel que décrit dans la Section 2.4.1.

4. Comme nous le verrons dans la Section 2.5, la mise en œuvre d'un quatrième arbre selon ce principe est inutile, pour la langue française tout du moins, car la stabilité d'une classe lexicale semble toujours garantie lorsque qu'on considère un contexte droit de trois mots.

5. la latence est ici définie comme le temps séparant la fin de la saisie d'un mot par l'utilisateur et sa synthèse

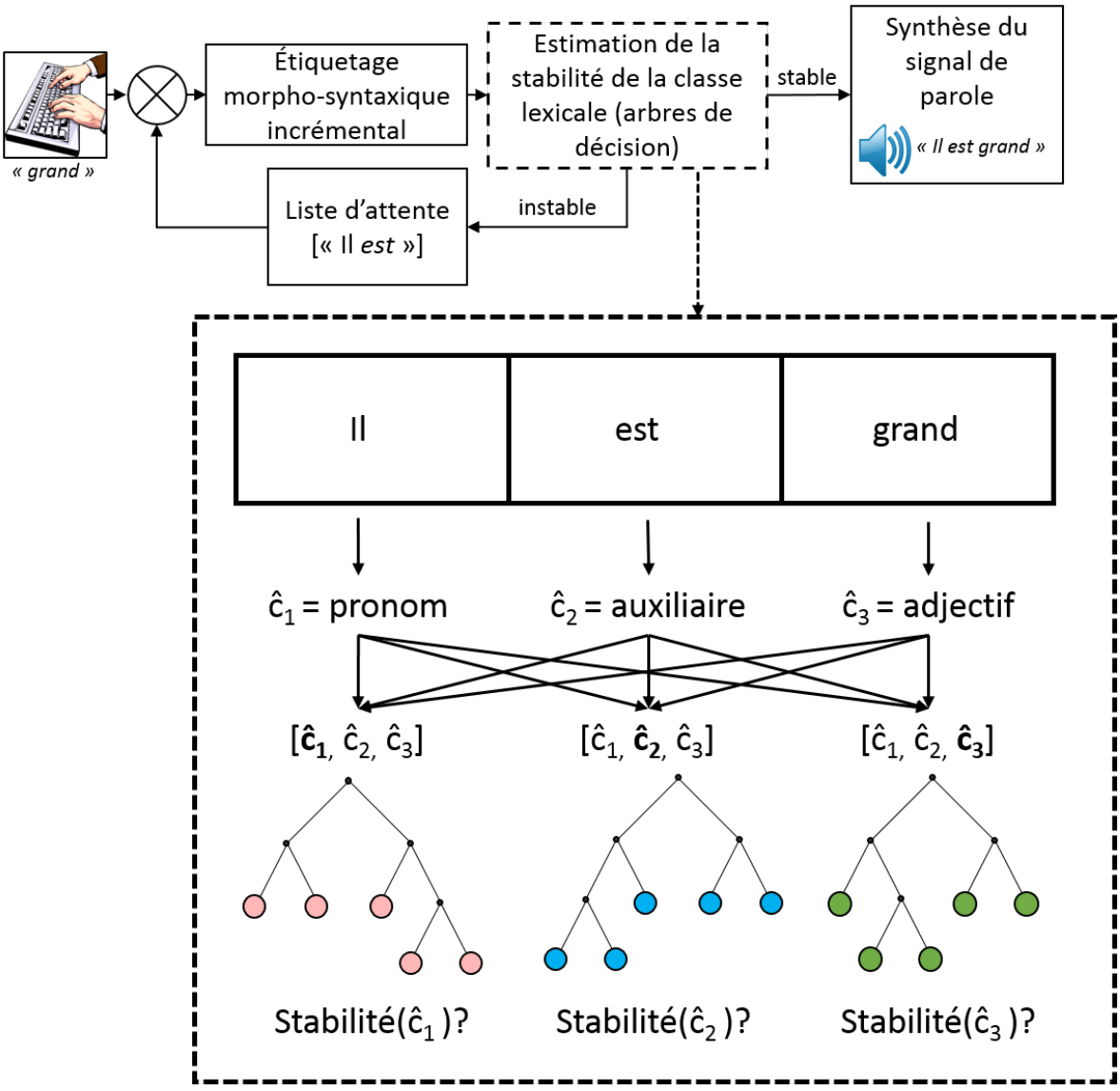


FIGURE 2.2 – Schéma-bloc résumant le principe de fonctionnement de l'algorithme d'analyse morpho-syntaxique proposé. La phrase tapée est "Il est grand et sympathique.". L'utilisateur a déjà entré les mots "Il est" dont les classes lexicales ont été jugées susceptibles de changer et ont donc été stockées dans une liste d'attente. Lors de la saisie du mot "grand", les arbres de décision estiment la stabilité des classes inférées pour les trois derniers mots. Si les trois classes sont estimées comme étant stables, alors les trois mots seront synthétisés ensemble.

À la saisie d'un nouveau mot m_i, la séquence de classes lexicales la plus probable est estimée à partir de ce mot, de son contexte gauche ($[m_0, ..., m_{i-1}]$), et d'un contexte droit de taille variable : nul pour le dernier mot entré, un mot de contexte droit pour l'avant dernier mot entré (m_i), et deux mots de contexte droit pour le précédent (m_{i-1} et m_i). À la différence de l'approche décrite précédemment, le calcul des densités de probabilité s'effectue ici non pas à l'aide de l'algorithme de Viterbi, mais de l'algorithme *Forward-Backward*.

L'algorithme *Forward Backward* se décompose en deux passes. La première, dite de *Forward*, détermine $\alpha_i(k) = P(c_k|m_i, m_{i-1}, ..., m_0)$, c'est-à-dire la probabilité d'être dans l'état c_k sachant le mot courant et son contexte gauche. Elle s'obtient à l'aide de la formule de récurrence suivante :

$$\alpha_i(k) = \sum_{j=1}^{K} \alpha_{i-1}(j) a_{jk} b_j(m_i) \tag{2.6}$$

La classe lexicale associée au dernier mot saisi m_i est naturellement définie comme celle maximisant cette probabilité *Forward*, tel que $\hat{c}_i = q_{r_i}$ avec $r_i = \text{argmax}_k\{\alpha_i(k)\}$.

A chaque saisie d'un nouveau mot m_i, la probabilité $P(c_g|m_0, ..., m_g, ..., m_i)$ associée à chaque mot m_g du contexte gauche de m_i (avec $g < i$) est également ré-estimée. Cette ré-estimation s'effectue en trois étapes : tout d'abord en calculant la probabilité *Forward* $\alpha_g(k)$ à l'aide de l'Équation 2.6, puis en calculant la probabilité dite *Backward* $P(c_g|m_g, m_{g+1}, ..., m_i)$ à l'aide de la formule de récurrence suivante :

$$\beta_g(k) = \sum_{j=1}^{K} \beta_{g+1}(j) a_{kj} b_j(m_g) \tag{2.7}$$

puis en multipliant les probabilités *Forward* et *Backward* telles que :

$$P(c_g|m_0, ..., m_g, ..., m_i) = \alpha_g(k)\beta_g(k) \tag{2.8}$$

Ici encore, la classe lexicale associée à chaque mot m_g du contexte gauche du mot m_i, est définie par $\hat{c}_g = q_{r_g}$ avec $r_g = \text{argmax}_k\{P(c_k|m_0, ..., m_g, ..., m_i)\}$. Cette implémentation basée sur l'algorithme *Forward-Backward* permet donc d'obtenir, pour chaque état, et pour chaque mot d'une séquence, les valeurs des probabilités a posteriori pour l'ensemble des classes. Rappelons que ces probabilités sont ici estimées de façon plus précise que dans l'algorithme de Viterbi, car elles intègrent les contributions de toutes les séquences d'états possibles. Ces probabilités – qui peuvent changer au fur et à mesure des variations de $\beta_g(k)$ –, associées à la séquence de classes $[\hat{c}_1, ..., \hat{c}_i]$, sont notées $[\hat{\gamma}_1, ..., \hat{\gamma}_i]$. Elles sont notamment exploitées dans la procédure d'apprentissage des arbres de décision qui est décrite à la section suivante.

2.4.3.2 Apprentissage des arbres de décision

Nous décrivons à présent la procédure d'apprentissage des trois arbres de décisions qui évaluent la stabilité d'une classe lexicale (décodée à l'aide de la procédure présentée dans la section précédente), en fonction de la taille du contexte droit considéré (aucun mot pour le premier arbre, un mot pour le second et deux mots pour le troisième). Pour chaque arbre, les variables d'entrée sont :

— $[\hat{c}_{i-2}, \hat{c}_{i-1}, \hat{c}_i]$ (les 3 dernières classes lexicales décodées, m_i est le dernier mot saisi)
— $[\hat{\gamma}_{i-2}, \hat{\gamma}_{i-1}, \hat{\gamma}_i]$ (les probabilités associées)

La variable en sortie de ces arbres de décision est un booléen indiquant si la classe lexicale estimée à partir du mot courant, de son contexte gauche, et d'un contexte droit de taille variable est identique à celle qui est estimée en considérant la phrase complète. Autrement dit, cette variable sera "Vraie" si la classe lexicale $\hat{c}_i$ estimée pour le mot m_i à partir de $[m_1, \ldots, m_i, \ldots, m_{i+L})]$ est identique à celle estimée à partir de $[m_1, \ldots, m_N]$, avec L la taille du contexte droit (et donc de l'arbre) considéré ($L \in \{0, 1, 2\}$), et N le nombre de mots dans la phrase d'apprentissage. Elle sera "Fausse" dans le cas contraire. Notons par ailleurs que l'on attribue à $\hat{c}_{-1}$ et $\hat{c}_0$ un marqueur explicite de "début de phrase" (BoS) auquel on associe une probabilité égale à 1.

Le Tableau 2.1 illustre la façon dont nous mettons en forme un texte annoté pour l'apprentissage des arbres de décision. L'apprentissage des arbres de décisions s'effectue sur un grand corpus de texte (détaillé à la Section 2.5.2), pour lequel la classe lexicale de chaque mot est connue, et mis en forme selon la procédure détaillée précédemment.

Plusieurs algorithmes peuvent être utilisés pour l'apprentissage de la structure et des paramètres d'un arbre de décision binaire. Nous utilisons ici une procédure standard basée sur l'algorithme proposé par (Breiman et al. 1984). Cette méthode consiste dans un premier temps à diviser de façon récursive et le plus efficacement possible les exemples de l'ensemble d'apprentissage en choisissant pour chaque nœud une question binaire. À chaque itération, si l'ensemble des éléments associés à une feuille n'appartiennent pas à la même classe, un nouveau critère de division est choisi. Dans le cadre de la méthode proposée, ce critère binaire peut être de deux formes :

— "$\hat{c}_k \in C$" ; avec $k \in \{i-2,\ i-1,\ i\}$ et C un ensemble de classes lexicales (par exemple $C = \{\text{Nom, Adjectif, Conjonction}\}$). C peut être réduit à un singleton.
— "$\hat{\gamma}_k \geqslant x$" ; avec $k \in \{i-2,\ i-1,\ i\}$ et $x \in [0, 1]$

Dans un second temps, l'arbre ainsi obtenu est élagué par minimisation d'une fonction de coût. Cet élagage se fait testant l'arbre sur un corpus de validation (disjoint du corpus d'apprentissage) en cherchant un compromis entre complexité de l'arbre et erreur sur le corpus de validation.

2.4.3.3 Algorithme complet d'analyse morpho-syntaxique incrémentale

Dans cette section, nous présentons l'algorithme complet d'analyse morpho-syntaxique incrémentale, qui cadence la synthèse des mots au fur et à mesure de leur saisie. La décision de synthétiser un mot est conditionnée par la stabilité de sa classe lexicale, estimée par l'ensemble des arbres de décisions. Notons, de plus, qu'un mot dont la classe lexicale n'est pas estimée comme étant stable verra sa propre synthèse retardée mais retardera également la synthèse des mots suivants (même si la classe lexicale de ceux-ci est estimée comme étant stable). L'approche proposée est décrite par l'Algorithme 2.

Dans la section suivante, nous décrivons le protocole expérimental mis en œuvre pour évaluer l'approche proposée.

TABLE 2.1 – Construction du corpus d'apprentissage des arbres de décision binaire pour l'estimation de la stabilité des classes lexicales inférées à l'aide de la méthode proposée.
Les classes estimées et leurs probabilités associées situées sur une même ligne sont utilisées pour les trois arbres. Les valeurs binaires en gras (**Vrai** ou **Faux**) sont les valeurs de sorties que l'on impose dans le cadre de l'apprentissage supervisé pour l'arbre correspondant (L = 2,1 ou 0).

		Ils	vont	en	vacances	à	la	mer	.
		Pp	Vrb	Pre	Nom	Pcn	Det	Nom	Pt
BoS 1	BoS 1	Pp 0.99 **Vrai**							
	BoS 1	Pp 0.99 **Vrai**	Vrb 0.98 **Vrai**						
		Pp 0.99 **Vrai**	Vrb 0.98 **Vrai**	Pp 0.14 **Faux**					
			Vrb 0.98 **Vrai**	Pre 0.85 **Vrai**	Nom 0.62 **Vrai**				
				Pre 0.85 **Vrai**	Adq 0.16 **Faux**	Nom 0.01 **Faux**			
					Nom 0.62 **Vrai**	Pre 0.21 **Faux**	Pp 0.05 **Faux**		
						Pcn 0.71 **Vrai**	Det 0.52 **Vrai**	Nom 0.49 **Vrai**	
							Det 0.52 **Vrai**	Nom 0.49 **Vrai**	Pt 0.23 **Vrai**

Data:

- Les trois derniers mots entrés $[m_{i-2}, m_{i-1}, m_i]$ et les classes lexicales inférées incrémentalement $[c_{t-2}, c_{t-1}, c_t]$ (voir Section 2.4.3.1)
- Une liste d'attente *waiting list* contenant les mots dont la classe lexicale n'a pas été estimée comme étant stable. En notant m_i le dernier mot tapé, *waiting list* peut potentiellement contenir $[m_{i-3}, m_{i-2}, m_{i-1}]$ (représenté sur la Figure 2.2).

```
if m_{t-3} is in waiting list then
    Synthesize(m_{t-3})
if m_{t-2} is in waiting list then
    if IsStable(c_{t-2}) (Contexte droit = 2 mots) then
        Synthesize(m_{t-2})
    else
        Put m_{t-2}, m_{t-1}, m_t in waiting list ;
        return ;
if m_{t-1} is in waiting list then
    if IsStable(c_{t-1}) (Contexte droit = 1 mot) then
        Synthesize(m_{t-1})
    else
        Put m_{t-1}, m_t in waiting list ;
        return ;
if m_t is in waiting list then
    if IsStable(c_t) (Contexte droit = 0 mot) then
        Synthesize(m_t)
    else
        Put m_t in waiting list ;
        return ;
```

Algorithm 2: Algorithme complet d'analyse morpho-syntaxique incrémental, basé sur l'estimation en ligne d'une classe lexicale, à l'aide d'un ensemble d'arbres de décision binaires.

2.5 Protocole expérimental

2.5.1 Analyseur morpho-syntaxique COMPOST

Nos expériences, menées en langue française, utilisent le système de traitement du langage naturel du synthétiseur *Text-to-Speech* COMPOST (Bailly et Alissali 1992), développé au GIPSA-lab. Par soucis de concision, les spécificités de ce système ne seront pas entièrement détaillées ici. Nous mentionnons simplement que COMPOST implémente une chaîne de TAL classique, avec les différents modules décrits en Section 2.2 (pré-traitement, analyseur morphologique, etc.). Nous nous focalisons ici sur l'analyseur morpho-syntaxique de COMPOST, qui utilise une approche de type 3-gram (telle que décrite à la Section 2.4.1). Cet analyseur a été modifié pour intégrer la procédure de décodage basée sur l'algorithme *Forward-Backward* décrit à la Section 2.4.3.1. La liste des classes lexicales utilisées par COMPOST est donnée au Tableau 2.2.

Table 2.2 – Liste des classes lexicales et de leurs abréviations utilisées par l'analyseur morpho-syntaxique employé dans cette étude.

Vrb	Verbe conjugué	Pre	Préposition
Npr	Nom propre	Pcn	Complément du nom
Vo	Vocatif	Adq	Adjectif qualificatif
Num	Numéral	Pps	Participe passé
Pp	Pronom personnel	Vrg	Virgule (ponctuation)
Det	Déterminant	Pnp	Pronom impersonnel (e.g. On)
Adv	Adverbe	Ne	Ne (négation)
Aux	Auxiliaire	Pas	Pas (négation)
Pt	Point (ponctuation)	Inf	Infinitif
Nom	Nom commun	Ppt	Participe présent
Cco	Conjonction de coordination	Ccs	conjonction de subordination
Pri	Pronom Interrogatif	Adi	Adjectif interrogatif
Prl	Pronom Relatif		

Le module de phonétisation de COMPOST utilise les arbres de décision proposés par (Pagel, Lenzo et Black 1998) avec de légères modifications : pour distinguer les homographes hétérophones, qui, pour la plupart différent principalement sur la classe lexicale (à l'exception des situations ne pouvant être désambiguïsées que grâce à l'analyse sémantique, voir Section 2.2.4), une entrée supplémentaire indiquant la classe lexicale du mot auquel appartient le phonème à synthétiser est utilisée. Ainsi, la terminaison "ent" d'un *verbe* du premier groupe (à la troisième personne du pluriel donc) sera muette alors que s'il s'agit d'un nom (par exemple, événement), elle sera prononcée [ɑ̃].

Enfin, la liaison, le fait de prononcer la consonne finale d'un mot quand il est suivi d'un mot commençant par une voyelle, est également très dépendante du contexte (la liaison va être faite dans "*Le plus grand ami de* Tom." et pas dans "*Le plus grand a mis* de l'eau dans

le verre.") et est calculée lors de l'analyse syntaxique. On génère ainsi des liaisons latentes (qui sont effectivement prononcées en fonction de décisions supra-lexicales, notamment de groupement prosodique ou syntaxique) qui sont également utilisées en entrée du module de phonétisation. Dans la méthode proposée, le liaison, lorsque celle ci est réalisée, est considérée comme faisant partie du mot suivant.

2.5.2 Corpus de données

Le corpus utilisé pour entraîner (et évaluer) les arbres de décision binaires de l'algorithme proposé est extrait de deux romans français : "Le tour du monde en 80 jours", écrit par Jules Verne, et "Notre-Dame de Paris", de Victor Hugo[6]. Le corpus se compose 20154 phrases, soit 290801 mots. Deux-tiers du corpus (soit 13436 phrases sélectionnées aléatoirement ou environ 193 000 mots) ont été alloués à l'entraînement des arbres de décision et le tiers restant (6718 phrase, soit 98000 mots) est utilisé pour évaluer les performances de l'estimateur de stabilité proposé. La Figure 2.3 permet de visualiser la distribution des classes lexicales (voir Tableau 2.2) sur le corpus. Enfin, les arbres de décision ont été entraînés en utilisant la méthode proposée par (Breiman et al. 1984). Le choix des tests binaires pour chaque nœud se fait par optimisation de l'indice de diversité de Gini (Gini 1912). L'élagage de l'arbre de classification se fait en minimisant un coût d'erreur de classification pour chaque observation à chaque feuille de l'arbre.

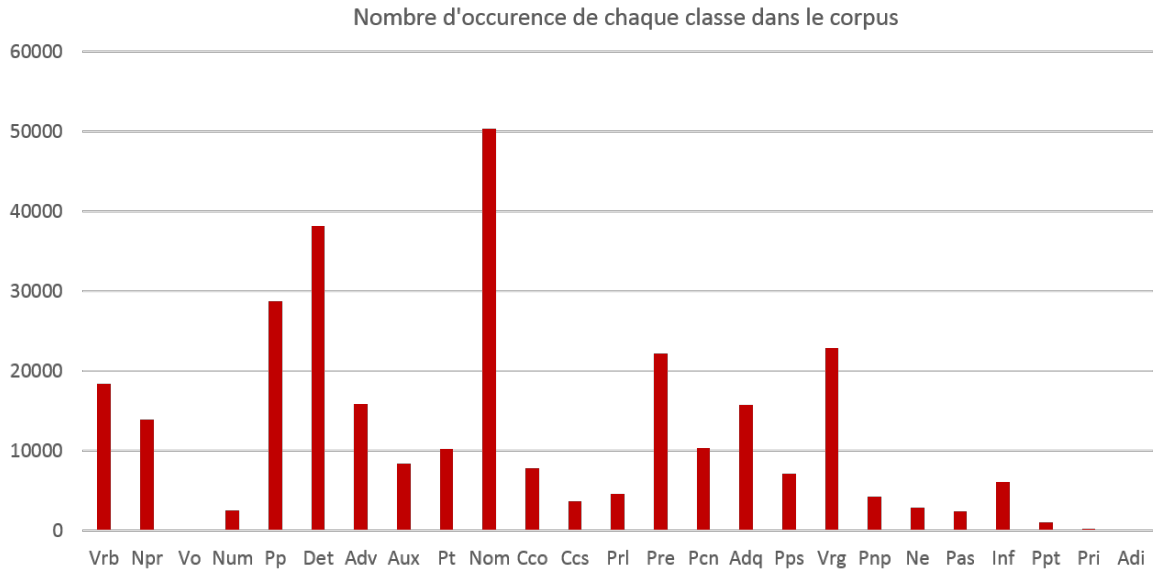


Figure 2.3 – Nombre d'occurrences de chaque classe sur le corpus total (apprentissage et test). Le Tableau 2.2 indique les correspondances entre abréviation et classe lexicale.

La Figure 2.4 illustre le sommet de l'arbre de classification estimant la stabilité de la classe

6. Il est à noter que le corpus issu du "Tour du monde en 80 jours" a été également utilisé pour l'entrainement des 3-gram, contrairement au corpus issu de "Notre dame de Paris".

lexicale du dernier mot tapé à partir des classes lexicales des trois mots précédents (avec leurs probabilités associées). Sur cette figure, on peut constater d'une part que les questions permettant de partitionner l'espace portent sur le dernier mot tapé (et pas sur les précédents pour les trois premiers niveaux de l'arbre estimant la stabilité de $\hat{c}_i$). D'autre part, on peut constater la présence d'une feuille (nœud terminal) donnant la stabilité de $\hat{c}_i$ si la probabilité a posteriori associée $\hat{\gamma}_i$ est supérieure à 0.93.

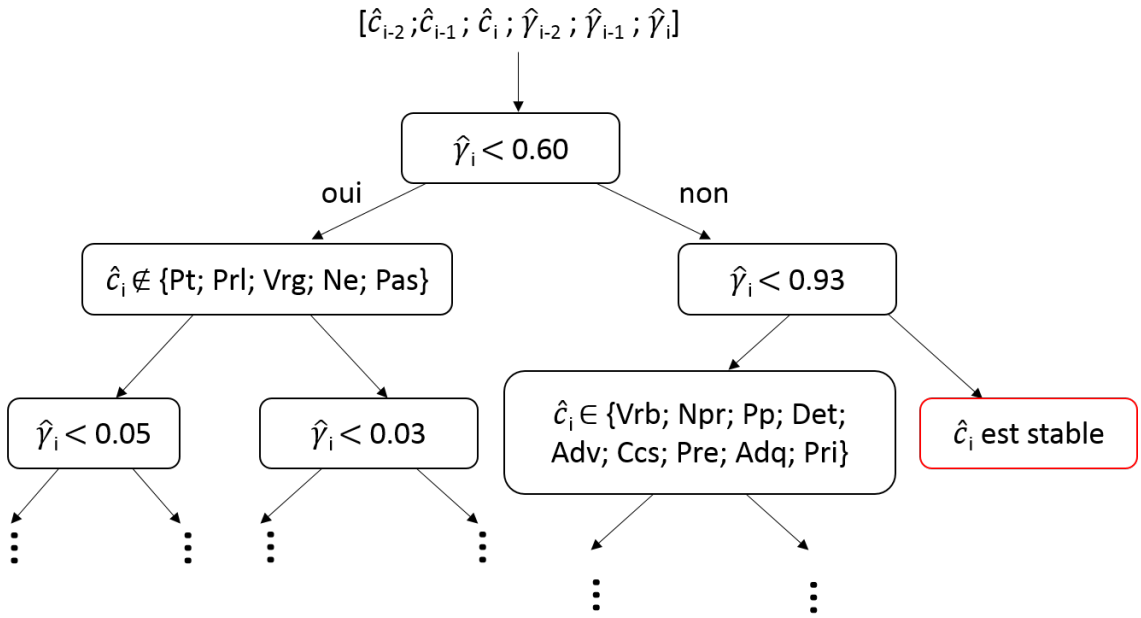


FIGURE 2.4 – Exemple d'arbre de classification permettant d'estimer la stabilité de la classe lexicale $\hat{c}_i$ du dernier mot tapé étant données les trois dernières classes lexicales décodées et de leur probabilité a posteriori $\hat{\gamma}_i$.

2.5.3 Évaluations objectives

L'évaluation de la méthode proposée s'effectue en deux étapes. Dans un premier temps, nous évaluons la performance de chacun des trois arbres, considéré indépendamment. Dans un second temps, nous évaluons la performance obtenue en cascadant les trois arbres tel que décrit dans l'Algorithme 2 (et dans l'exemple présenté en début de Section 2.4.2).

2.5.3.1 Métriques

Les performances des arbres de classification ont été évaluées en recensant le nombre de Vrais Positifs (VP), de Vrais Négatifs (VN), de Faux Négatifs (FN) et de Faux Positifs (FP). Ces catégories correspondent aux situations suivantes dans le cadre de l'estimation de la stabilité des classes lexicales :

— VP : l'algorithme estime à raison que la classe lexicale est stable, c'est-à-dire qu'elle ne va pas changer en ajoutant plus de contexte droit que celui déjà considéré (0, 1, ou 2 mots en fonction de l'arbre). Le déclenchement de la synthèse est donc légitime.
— VN : l'algorithme estime à raison que la classe lexicale est instable et que sa synthèse doit être retardée (le mot est donc placé dans la liste d'attente en vue d'une ré-analyse ultérieure)
— FP : l'algorithme estime à tort que la classe lexicale ne va pas changer en ajoutant plus de contexte droit que celui déjà considéré. La synthèse est alors déclenchée avec une mauvaise classe lexicale et donc potentiellement une mauvaise phonétisation.
— FN : l'algorithme estime à tort que la classe lexicale est instable. La synthèse a donc été inutilement retardée.

Au cours de ces évaluations objectives, pour un mot donné, on considère comme vérité de terrain la classe lexicale donnée par l'analyseur morpho-syntaxique lors de l'analyse de la phrase entière.

Différentes mesures permettent de qualifier les performances des arbres de décision :
On définit la précision (comprise entre 0 et 1) :

$$Acc = \frac{VP + VN}{VP + FP + FN + VN}$$

comme le taux de bonnes réponses données par l'algorithme.
Le calcul de la précision seule permet d'avoir une idée des performances des arbres de décision binaires utilisés ici. On peut cependant également s'intéresser au calcul de la sensibilité et de la spécificité des arbres de décision pour une interprétation plus complète de leur capacité à évaluer la stabilité de la classe lexicale d'un mot.

La sensibilité d'un test de décision mesure sa capacité à donner un résultat positif lorsque l'hypothèse est vérifiée. Elle s'oppose à la spécificité d'un test de décision, qui mesure sa capacité à donner un résultat négatif lorsque l'hypothèse n'est pas vérifiée. Sensibilité et spécificité se calculent de la manière suivante :

$$Sen = \frac{VP}{VP + FN}$$

$$Spe = \frac{VN}{VN + FP}$$

Notons que dans le cadre qui nous intéresse, celui de l'estimation de la stabilité d'une classe lexicale, un arbre de décision possédant une grande sensibilité aura tendance à déclencher la synthèse d'un mot même si sa classe lexicale a été incorrectement estimée. À l'inverse, un classifieur très spécifique aura plutôt tendance à retarder la synthèse inutilement.

2.6 Résultats et Discussion

La Figure 2.5 donne les performances des arbres de décision binaires considérés indépendamment.

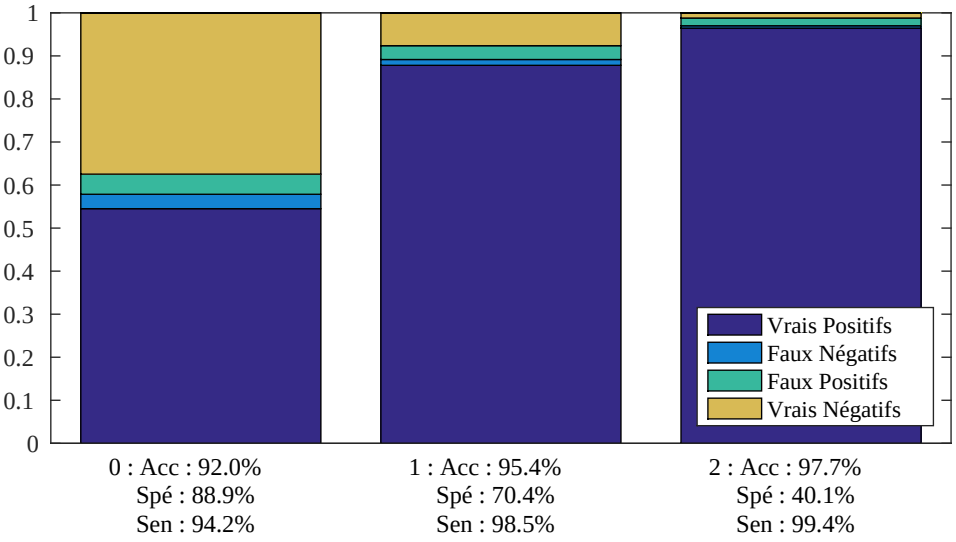


FIGURE 2.5 – Évaluation objective des arbres de décision binaire (considérés indépendamment) estimant la stabilité d'une classe lexicale en fonction de la taille du contexte droit considéré (de gauche à droite : 0, 1 et 2 mots).

Dans un premier temps, ces résultats nous permettent de discuter de la performance "brute" d'un analyseur morpho-syntaxique en contexte incrémental. Nous constatons qu'avec aucun contexte droit, la classe lexicale est correctement estimée dans 58% (VP+FN). De façon attendue, ce nombre augmente en fonction de la taille du contexte droit considéré : 89% avec un mot et 97% avec deux mots. Aussi, nous constatons que l'analyse morpho-syntaxique d'un mot m_i est quasiment aussi précise en considérant deux mots de contexte droit, qu'en considérant tout le reste de la phrase. Autrement dit, une analyse incrémentale avec un retard fixe de deux mots est quasiment aussi précise que l'analyse de la phrase complète (notre approche vise une latence encore inférieure).

Dans un second temps, nous discutons de la capacité des arbres de décision à évaluer en ligne la stabilité de la classe lexicale décodée. Le premier arbre de décision, qui ne considère aucun contexte droit, détermine correctement dans 92% des cas (VN+VF) la stabilité ou l'instabilité d'une classe lexicale : dans 55% des cas (VP) la synthèse est déclenchée sans risque, et dans 37% des cas (VN), elle est retardée à raison. À ce stade, on constate 5% de déclenchements trop soudain de la synthèse (FP) et 8% de retards inutiles (FN).

De façon attendue, ces performances s'améliorent en fonction de la taille du contexte droit considéré. Avec un mot de contexte droit, la performance globale (VP+VN) passe à $95,4\%$, le pourcentage de FP n'est plus que de 3%, celui de FN de 1%. Avec deux mots, on obtient une performance globale de $97,7\%$ et des pourcentages de FP (resp. FN) de 2% (resp. 1%) seulement [7].

Rappelons que la spécificité a trait aux mots qui devraient être mis en attente mais qui

7. ceci justifie notamment l'inutilité d'un 4ème arbre de décision prenant en compte trois mots de contexte droit, pour la langue française tout du moins.

sont synthétisés (donc principalement liés aux Faux Positifs). Nous pouvons constater que plus le nombre de mots dans le contexte droit augmente, moins le test que l'on réalise est spécifique : avec aucun contexte droit, $88,9\%$ des mots qui devraient être ré-analysés avec un mot supplémentaire dans le contexte droit sont effectivement placés dans la liste d'attente. Cette proportion descend à $70,4\%$ avec un mot de contexte droit puis à $40,1\%$ avec deux mots dans le contexte droit. Ces résultats nous montrent que le nombre de classes instables à détecter diminuant, les arbres de classification sont de moins en moins performants pour les détecter.

La sensibilité en revanche, passe de $94,2\%$ pour aucun contexte droit à $99,4\%$ pour deux mots dans le contexte droit. La sensibilité représentant la capacité à détecter un mot dont la classe lexicale a été correctement estimée, ces résultats confirment l'hypothèse qu'une latence maximale de 2 mots est suffisante.

Nous décrivons à présent les résultats obtenus lorsque nous cascadons les trois arbres de décision, tel que décrit dans l'Algorithme 2. Nous évaluons donc ici uniquement le taux de VP (mots synthétisés dont la classe lexicale est jugée stable), de FP (mots synthétisés avec une classe lexicale incorrecte), et le nombre de mots mis dans la liste d'attente par chacun des trois arbres et envoyé à l'arbre suivant. Les résultats obtenus sont présentés à la Figure 2.6.

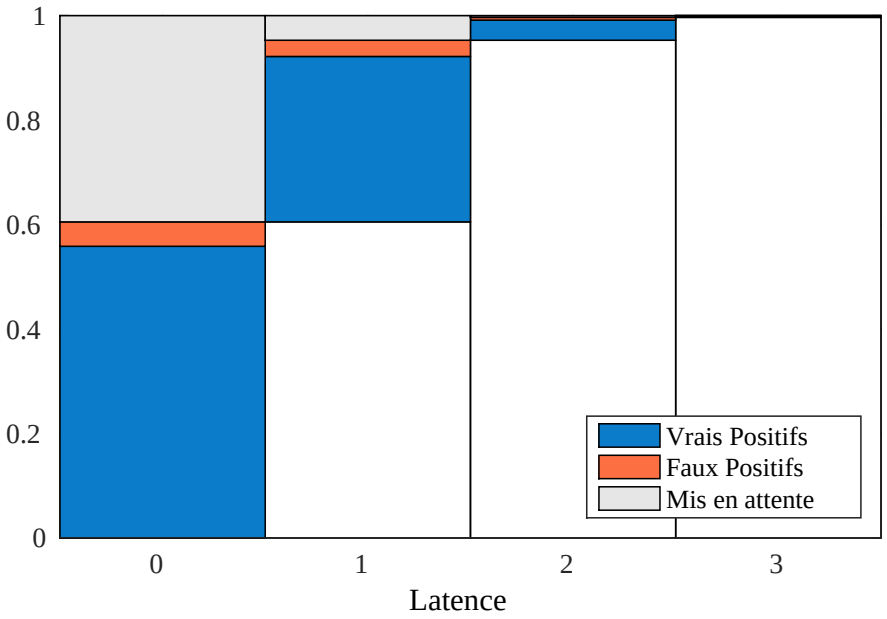


FIGURE 2.6 – Évaluation objective des arbres de décision binaires cascadés : estimation de la stabilité d'une classe lexicale en fonction de la taille du contexte droit considéré (de gauche à droite : 0, 1 ou 2 mots).

Ainsi, environ 60% des mots sont synthétisés dès leur saisie (avec une classe lexicale qui ne changera a priori pas si on considère un contexte droit plus important). Dans 35% des cas, un mot de contexte droit est nécessaire, et deux mots dans 4% des cas. Enfin, dans une très

faible proportion de cas, 0,3% d'entre eux, la synthèse d'un mot est déclenchée après obtention de trois mots supplémentaires (cela correspond au cas où les trois arbres ont jugé la classe instable).

En pratique, la séquence "Nom commun précédé d'un déterminant lui même précédé d'une préposition (donc la séquence [Préposition – Déterminant – Nom], en considérant la stabilité du Nom) apparaît 9880 fois dans le corpus d'apprentissage. Celle-ci est calculée comme étant stable dans 9014 cas (soit 91% des fois où cette séquence apparaît). À l'inverse, la synthèse d'un nom propre considéré dans la séquence [Nom propre – Virgule – Pronom Interrogatif] et situé en début de phrase ne sera déclenchée qu'après l'ajout du 3^{eme} mot.

Nous pouvons également calculer un taux d'erreur d'estimation de classe lexicale en fonction du nombre de mots dans le contexte droit ainsi qu'un taux d'erreur global, pour l'ensemble des mots synthétisés. On calcule le taux d'erreur pour un nombre de mots dans le contexte droit donné de la manière suivante :

$$taux\ d'erreur = \frac{FP}{FP + VP}$$

et le taux d'erreur pour l'ensemble du système de la manière suivante :

$$taux\ d'erreur_{global} = \frac{\sum_i FP_i}{\sum_i FP_i + \sum_i VP_i}$$

avec FP_i et VP_i les Faux et Vrai Positifs pour l'arbre i, $i = 0$, 1, 2 ou 3.

Avec 0 mot dans le contexte droit, le taux d'erreur s'élève à 6%. Cette valeur augmente avec le nombre de mots dans le contexte droit. Avec 1 mot dans le contexte droit, l'arbre de classification considère la classe lexicale estimée incrémentalement comme correcte avec un taux d'erreur de $8,5\%$. Enfin, avec 2 mots dans le contexte droit, ce taux d'erreur atteint 11%. Le taux d'erreur global pour l'estimateur de stabilité est de $7,5\%$. Ces résultats nous montrent qu'en proposant une latence adaptative lors d'une analyse morpho-syntaxique incrémentale, il est possible d'estimer correctement la classe lexicale d'un mot avec $92,5\%$ de précision. Enfin, ces données nous permettent également de calculer une durée d'attente moyenne (en nombre de mots) de 1,4 mots.

2.7 Conclusions et perspectives

Dans ce chapitre, nous avons rappelé les principes généraux de fonctionnement d'une chaîne de traitement automatique de la langue naturelle (TAL) pour la synthèse de la parole à partir du texte. Nous avons identifié les verrous technologiques à lever pour effectuer les différents traitements linguistiques dans un contexte incrémental, pour lequel une phrase doit être analysée au fur et à mesure de sa saisie. Parmi ces traitements, nous nous sommes intéressé plus spécifiquement à l'analyse morpho-syntaxique. Nous avons proposé une nouvelle approche pour effectuer une analyse morpho-syntaxique robuste dans un contexte incrémental. Notre méthode s'appuie sur une estimation en ligne du contexte droit minimum à considérer pour garantir une analyse correcte du dernier mot saisi. Cette estimation s'appuie sur les classes

lexicales attribuées aux trois derniers mots saisis (incluant le mot courant) et sur les probabilités associées à ces classes. Ces probabilités sont obtenues à l'aide d'une adaptation simple d'un analyseur basé sur une modélisation de type n-gram (Section 2.4.1), à l'aide de l'algorithme *Forward Backward*. L'estimation en ligne de la stabilité d'une classe lexicale en fonction du contexte droit considéré est obtenu à l'aide d'un ensemble de trois arbres de décision dont les paramètres sont estimés par apprentissage automatique supervisé sur un large corpus de texte.

La méthode proposée permet d'évaluer correctement la classe lexicale d'un mot dans un contexte incrémental avec un taux d'erreur de $92,5\%$ et une latence moyenne de 1,4 mots.

Afin d'améliorer la précision de la méthode d'analyse morpho-syntaxique à latence adaptative proposée, de futurs travaux pourraient consister à utiliser non pas une seule classe lexicale par mot mais les n meilleures classes lexicales inférées à l'aide de l'algorithme Forward Backward. Une autre perspective est l'évaluation de cette approche dans d'autres langues, comme l'allemand ou l'anglais, pour lesquels nous pouvons supposer que l'analyse syntaxique nécessite l'exploitation d'une fenêtre contextuelle potentiellement plus large en raison d'ambiguïtés lexicales plus nombreuses (le mot anglais *like*, par exemple, peut être associé à plus de 7 classes lexicales différentes, selon le contexte). Enfin, notons que dans ce travail de thèse, ce principe d'estimation en ligne de la stabilité dans des modules de TAL en contexte incrémental, est appliqué uniquement à l'analyse morpho-syntaxique. Cependant, il semble possible de le transposer aux autres modules de la chaîne de TAL exploitant également des informations contextuelles pour le traitement linguistique d'un mot. La validation de l'approche proposée pour l'ensemble des modules de la chaîne de TAL est une des perspectives principales de ce travail.

Dans le chapitre suivant, nous nous intéressons à l'adaptation du module de synthèse sonore à la synthèse incrémentale.

Chapitre 3

Synthèse sonore paramétrique incrémentale

Sommaire

3.1 Introduction

Le schéma général d'un synthétiseur de parole à partir du texte est rappelé à la Figure 3.1. Ce chapitre porte sur l'adaptation du second bloc, à savoir la génération de la forme d'onde, à notre contexte de synthèse incrémentale. Nous rappelons que ce module convertit une séquence de descripteurs linguistiques décrivant le contenu phonétique et syntaxique issue du module de TAL en un signal sonore.

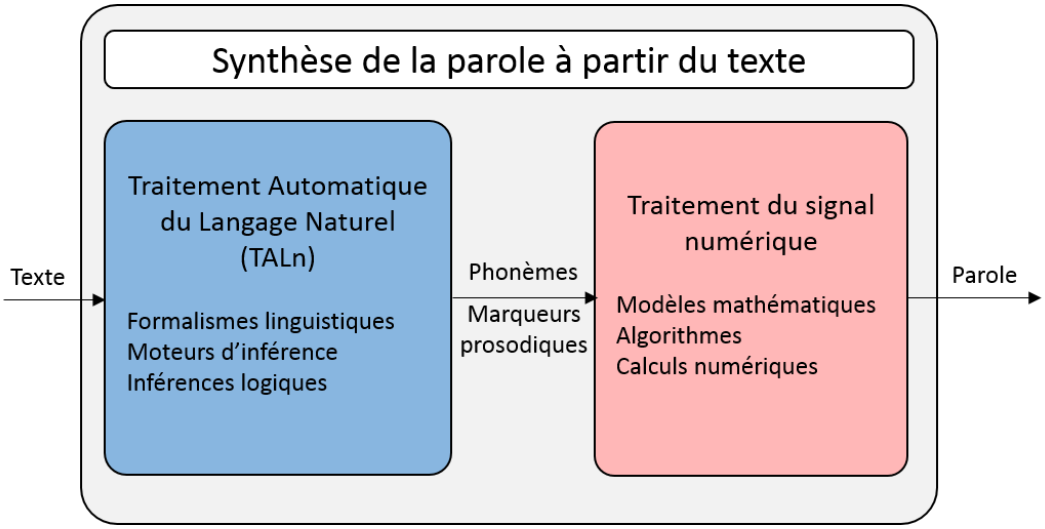


FIGURE 3.1 – Diagramme fonctionnel d'un système de synthèse TTS inspiré de (BOITE et al. 2000).

Ce chapitre s'organise de la façon suivante. Dans un premier temps (Section 3.2), nous rappelons les différentes approches envisageables pour cette étape de synthèse sonore. Nous nous focaliserons ensuite, en Section 3.3, sur l'approche dite de "synthèse paramétrique", et plus spécifiquement sur la synthèse paramétrique par Modèles de Markov Cachés (on parlera alors de "synthèse par HMM"). Nous dégagerons, en Section 3.4, les verrous à lever pour une utilisation de la synthèse par HMM en contexte incrémental et présenterons les premières pistes proposées dans la littérature. Puis nous détaillerons, en Section 3.4.2, la stratégie que nous proposons dans cette thèse. Enfin, dans la Section 3.5, nous présenterons le protocole expérimental mis en œuvre pour évaluer notre méthode et nous discuterons des résultats obtenus.

3.2 Synthèse vocale, état de l'art

Les synthétiseurs de parole peuvent être regroupés en trois classes qui ont émergé au fil du temps avec l'amélioration des capacités de stockage et de calcul des ordinateurs : la synthèse par règles, la synthèse par corpus et la synthèse paramétrique statistique. Cette brève description des différents systèmes de synthèse et leurs exemples s'appuie en partie sur (D'ALESSANDRO 2001) ainsi que sur (BOITE et al. 2000).

3.2.1 Synthèse par règles

La synthèse par règles consiste à mettre au point, de façon experte, un ensemble de valeurs cibles et de trajectoires pour modéliser les paramètres représentant le signal acoustique associé à une séquence phonétique. Un exemple de synthèse par règles exploitant les représentations paramétriques est la synthèse par formants. Les formants sont des bandes de fréquence présentant un maximum local d'énergie. Les paramètres d'un synthétiseur par formants sont, pour chacun des k premiers formants ($k = 3$ ou 4 en pratique), et pour chacun des phonème p, la fréquence centrale f_{p_k}, la largeur de bande B_{p_k}, et les règles de transition permettant de passer de $\{f_{p_K},\ B_{p_K}\}$ à $\{f_{p+1_K},\ B_{p+1_K}\}$. Ces valeurs, déterminées de façon experte, sont généralement stockées dans une table et utilisées lors de la synthèse. Un exemple de synthèse par formants pour la synthèse de l'anglais est décrit dans (Klatt 1987). Enfin, (Olaszy, Gordos et Németh 1992) proposent une méthode consistant non pas à relier les valeurs cibles par des fonctions de transition mais à ordonnancer des unités acoustiques élémentaires (*Acoustic Building Units*). Pour le français, des exemples de signaux de synthèse obtenus à l'aide de synthétiseurs par règles peuvent être trouvés dans (d'Alessandro 2001).

3.2.2 Synthèses par concaténation d'unités

Dans ce type de système, le signal de parole est construit par concaténation de segments audio pré-enregistrés, sélectionnés automatiquement dans une base de données. La nature de ces segments et l'algorithme de sélection permettent de classer les différents systèmes. Dans les systèmes de synthèse par diphones, on utilise des unités allant de la partie stable d'un phone à la partie stable du phone suivant. Ce choix vise à limiter la distorsion introduite à l'endroit de la concaténation (qui s'effectue sur une portion spectralement stable). La sélection des unités s'effectue simplement à partir de la transcription phonétique.

Avec l'augmentation de la mémoire des ordinateurs, vers la fin des années 80, (Sagisaka 1988) propose de stocker plusieurs exemplaires pour une même classe d'unité. Chaque unité se voit alors multi-représentée dans des contextes linguistiques et prosodiques variés. On parle alors de synthèse par sélection d'unités, car la taille et la richesse de la base de données et l'algorithme de sélection des unités vont jouer un rôle majeur dans la qualité de la voix de synthèse. Disposer d'un grand nombre d'exemplaires par unité, va notamment permettre de minimiser les opérations de modification du signal et les distorsions qu'elles engendrent. Par ailleurs, l'utilisation d'unités de taille variable, allant du phone, au polyphone (en passant par le diphone), voire au mot, permet de limiter le nombre de concaténations nécessaires lors de la synthèse (voir Taylor et Black 1999). Dans les systèmes actuels (commerciaux comme ouverts), la base de données contient plusieurs heures de parole étiquetée (l'étiquetage faisant souvent l'objet d'une revérification par un opérateur humain). Cette utilisation d'unités audio permet aux synthétiseurs par sélection d'unités d'atteindre une excellente intelligibilité et une qualité de synthèse très proche de ce que peut produire un humain. Parmi les différentes approches proposées dans la littérature pour la sélection des unités à partir des descripteurs symboliques fournit par le TAL, on citera notamment l'algorithme de Viterbi (déjà décrit dans

le chapitre précédent pour l'analyse syntaxique). Il est ici utilisé pour déterminer la séquence d'unités qui minimise conjointement un *coût de sélection* se rapportant à la cible linguistique, et un *coût de concaténation* se rapportant à la transition entre deux unités sélectionnées. La synthèse par concaténation d'unités n'étant pas au cœur de ce travail, nous renvoyons le lecteur vers (Hunt et Black 1996) pour plus de détails.

3.2.3 Synthèse paramétrique statistique

La synthèse paramétrique statistique s'appuie, comme son nom l'indique, sur une représentation paramétrique du signal de parole. Cependant, les cibles et les transitions acoustiques ne sont, ici, pas fixées *a priori* par des experts, mais apprises automatiquement à partir de grandes bases de données (similaires à celles utilisées pour la synthèse par sélection d'unités). L'approche classique, développée notamment par le groupe de recherche HTS (Tokuda et al. 1999), est basée sur les modèles de Markov cachés. Cette approche est au cœur de ce travail de thèse et sera décrite en détail dans la Section 3.3. Parmi les avantages de l'approche par modèles statistiques, on citera notamment la possibilité de créer une nouvelle voix de synthèse par adaptation des modèles à partir d'un corpus limité d'enregistrements d'un nouveau locuteur (Masuko et al. 1997, Tamura et al. 2001). On citera également la possibilité de créer une nouvelle voix de synthèse en contrôlant certaines propriétés acoustiques comme la qualité vocale, par interpolation d'une ou plusieurs autres voix de synthèse existantes (Yoshimura et al. 2000).

Une autre approche pour la synthèse sonore paramétrique est d'exploiter un réseau de neurones artificiels (Karaali, Corrigan et Gerson 1996). Récemment, et en ligne avec les récents travaux sur le *deep learning*, de nouvelles implémentations basées sur les réseaux de neurones profonds et/ou récurrents (basés par exemple sur l'architecture *Long Short Term Memory* ou LSTM) ont vu le jour (Zen, Senior et Schuster 2013, Zen et Sak 2015). Les approches par réseaux de neurones permettent par exemple d'extraire automatiquement des caractéristiques du signal de parole provenant de différents locuteurs (Fan et al. 2015) ou même de différentes langues (Fan et al. 2016) grâce à des nœuds d'étranglements au sein du réseau de neurones.

3.3 Synthèse paramétrique par HMM

3.3.1 Principe général

Nous rappelons ici le principe général d'un système de synthèse TTS par HMM, tel que proposé par le groupe de travail HTS (qui développe le *toolkit* du même nom HTS 2000). Cette description s'appuie notamment sur (Tokuda et al. 2013). Un schéma général de fonctionnement du système HTS est proposé à la Figure 3.2.

Comme mentionné précédemment, la mise en œuvre d'un système de synthèse par HMM fait appel à une étape préalable d'apprentissage. Lors de cette étape, le contenu acoustique

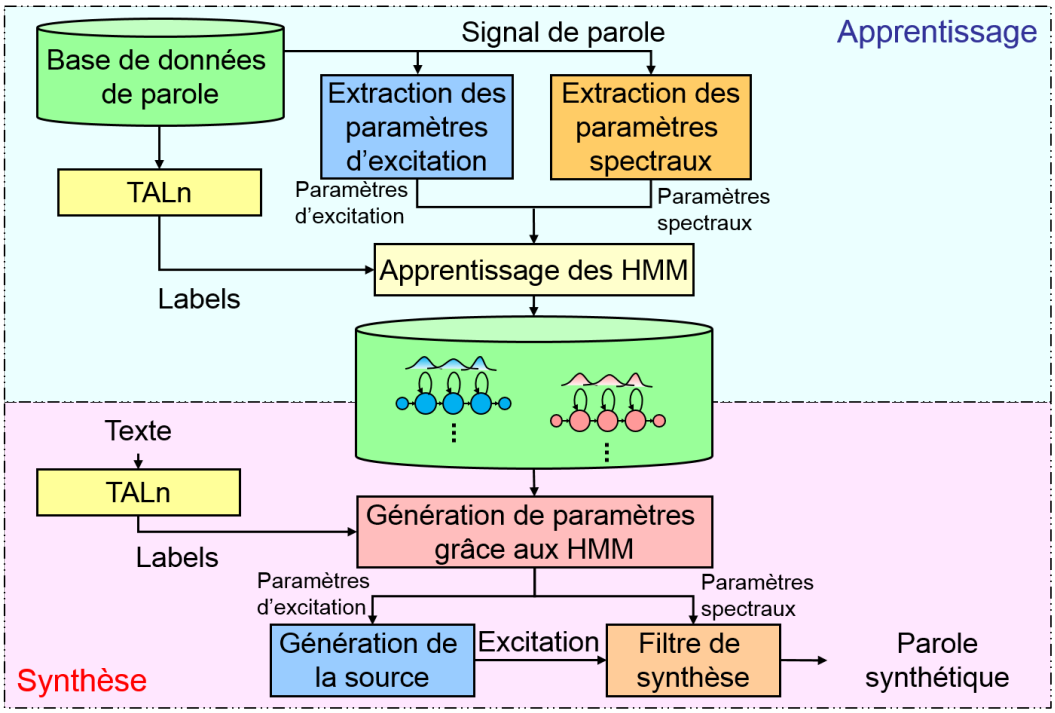


FIGURE 3.2 – Vue d'ensemble de la synthèse par HMM : Entraînement et synthèse. Figure inspirée de (TOKUDA et al. 2013). La partie supérieure du schéma résume la tâche d'apprentissage des HMM tandis que la partie inférieur schématise le processus de synthèse de parole.

du signal de parole est modélisé, pour chaque classe phonétique (et comme nous le verrons ultérieurement, pour chaque contexte segmental et suprasegmental), par un modèle de Markov Caché ou HMM (pour *Hidden Markov Model*). Les HMM sont des modèles génératifs, utilisés depuis de nombreuses années en reconnaissance automatique de la parole, et connus pour bien rendre compte la structure temporelle de la parole. Un rappel théorique sur les HMM est présenté à la Section 3.3.5. Les spécificités de l'utilisation des HMM pour la synthèse de la parole seront décrites à la Section 3.3.6. La synthèse par HMM s'appuie sur une représentation paramétrique du signal de parole, c'est-à-dire un encodage de son contenu acoustique en une séquence d'observations (vecteur de paramètres continus), que nous appelons dans la suite "observation acoustique". Plusieurs approches peuvent être envisagées pour le paramétrage acoustique. La plus classique (telle qu'utilisée dans HTS) s'appuie sur une modélisation "source-filtre", qui sous-tend que le signal de parole peut être décrit comme le filtrage d'un signal d'excitation (représentant l'activité glottique ou les constrictions supra-glottiques) par un filtre dont la réponse en fréquence varie au cours du temps, représentant les cavités supra-glottiques. Le signal de parole est donc décrit par deux ensembles de paramètres, les paramètres d'excitation (pour la source) et les paramètres spectraux (pour le filtre). Dans ce travail, nous décrivons une autre approche, basée sur une modélisation pleine-bande du spectre de la parole, à l'aide du modèle "harmonique plus bruit" (Stylianou 1996). En phase de synthèse, une séquence d'observations acoustiques est estimée à partir des modèles HMM entraînés lors de la phase d'apprentissage, et de la cible linguistique issue du module de TAL. L'algorithme d'inférence des observations acoustiques sera décrit à la Section 3.3.7.2. Enfin, la parole de synthèse est générée à l'aide d'un vocodeur, transformant une séquence d'observations acoustiques en une séquence d'échantillons audio. Les vocodeurs associés à la décomposition mel-cepstrale et à la modélisation harmonique plus bruit sont décrits ci-après.

3.3.2 Représentation paramétrique du signal de parole

3.3.2.1 Modélisation source-filtre de la production de la parole

Dans cette section, nous rappelons brièvement le principe général de la production de la parole, ainsi que le modèle source-filtre.

La production de la parole repose sur la modulation d'un flux d'air expulsé par les poumons dans les différentes cavités qui composent l'appareil vocal : la glotte, où se situe les plis vocaux, et les cavités orales (buccales et nasales) dont la géométrie, et donc les propriétés acoustiques varient en fonction des mouvements des articulateurs, principalement la langue, le voile du palais, la mâchoire et les lèvres.

Au niveau glottique, on distingue la production de sons voisés et de sons non-voisés. Lors de la production de sons non voisés, les plis vocaux sont écartés et le flux d'air pulmonaire n'est pas modifié. En revanche, lors de la production de sons voisés, les plis vocaux effectue un processus cyclique d'abduction (décollement des plis) et d'adduction (rapprochement des plis). La fréquence de ce cycle correspond à la fréquence fondamentale f_0 du signal de parole.

L'air pulmonaire, ainsi mis en forme au niveau de la glotte, vient ensuite résonner dans

le conduit vocal. Cette mise en résonance est un processus de filtrage de l'onde de débit d'air pulmonaire. Ainsi, l'appareil vocal, et donc le processus de production de la parole, est classiquement décrit comme un système source-filtre. Une vue schématique de cette théorie est rappelée à la Figure 3.3. Le modèle source-filtre de la production de la parole est à la base des techniques de paramétrage acoustique du signal de parole dans le cadre de la synthèse paramétrique.

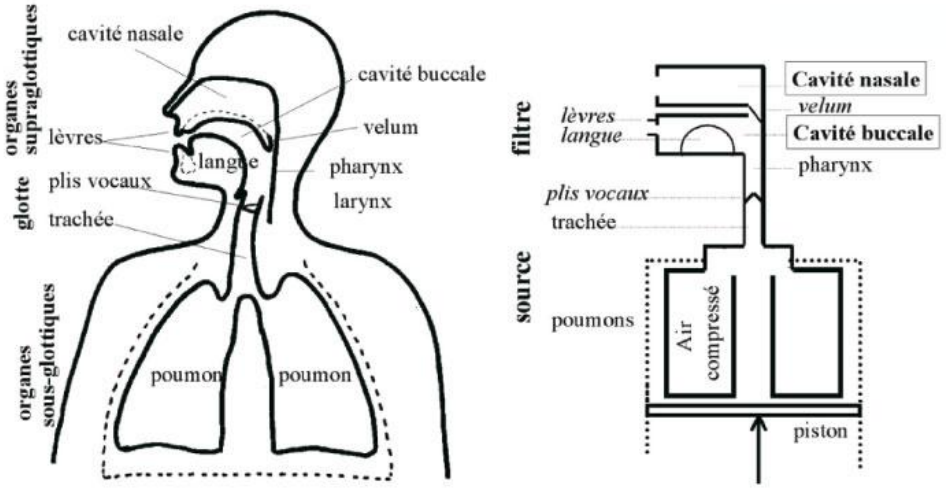


FIGURE 3.3 – Modélisation source-filtre de la production de la parole. À gauche, une représentation anatomique du conduit vocal identifiant les principaux articulateurs. À droite, une représentation mécanique schématique du conduit vocal. Extrait de (VAISSIÈRE 2011).

3.3.3 Analyse Mel-Cepstrale

L'approche classique utilisée pour le paramétrage du signal de parole en synthèse par HMM s'appuie sur une décomposition source-filtre, telle que décrite précédemment. Il s'agit d'encoder le signal de parole par deux jeux de paramètres distincts : les différentes sources d'excitation (glotte, bruits de constriction), c'est-à-dire le signal d'excitation, et le filtre, c'est-à-dire la fonction de transfert du conduit vocal. Plusieurs modèles peuvent être envisagés pour paramétrer cette enveloppe spectrale parmi lesquels, le modèle LPC (Linear Predictive Coding) et le modèle Mel-Cepstre complexe tel que proposé par (IMAI 1983), utilisé par défaut dans HTS.

Le spectre complexe $H(z)$ d'une trame de signal est ici modélisé par l'équation suivante :

$$
\begin{aligned}
H(z) &= \exp \sum_{m=0}^{M} c(m)\hat{z}^{-m} \\
\text{Avec :} \quad \hat{z}^{-1} &= \Psi_\alpha(z) = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}} \\
\text{et} \quad 0 &\leqslant \alpha \leqslant 1
\end{aligned}
\tag{3.1}
$$

L'enveloppe spectrale est donc ici paramétrée par les coefficients dit mel-cepstraux complexes $\{c_m\}_{m\in[1,M]}$, que nous noterons par la suite coefficients MGC. La fonction $\Psi_\alpha(z)$ est un filtre déphaseur qui permet d'approximer l'échelle de Mel (pour un signal échantillonné à 16kHz, on prend typiquement $\alpha = 0.42$). Il s'agit d'une échelle logarithmique approximant l'échelle de perception acoustique humaine (il est plus facile de distinguer deux fréquences proches lorsqu'elles sont basses que lorsqu'elles sont hautes).

De façon similaire au filtre LPC , (Imai 1983) propose une méthode pour dériver un filtre numérique à partir d'un jeu de coefficients MGC ; il s'agit du filtre MLSA, pour *Mel Log Spectral Approximation*. Il est donc possible de synthétiser un signal de parole à partir d'une séquence de vecteurs de coefficients MGC, et d'un signal d'excitation.

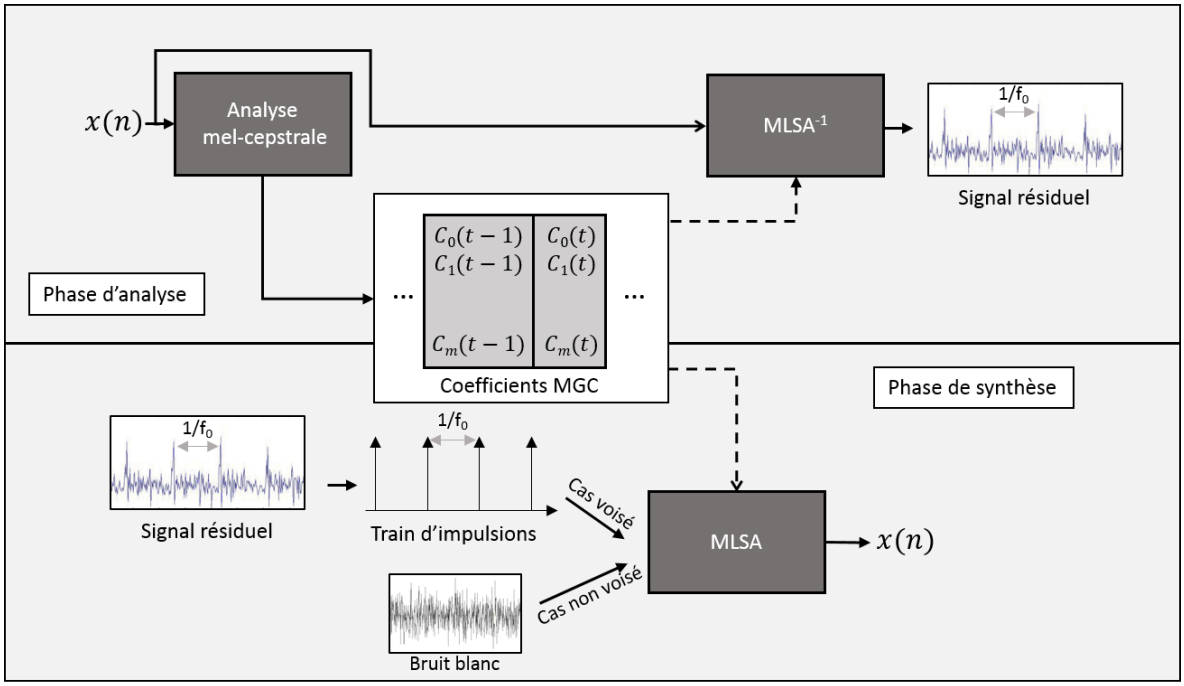


FIGURE 3.4 – Schéma bloc représentant le fonctionnement de l'analyse et de la synthèse d'un signal de parole dans le cadre de la modélisation Mel-Cepstrale.

Ce signal d'excitation peut être obtenu par filtrage MLSA inverse du signal original, comme illustré à la Figure 3.4. Plusieurs approches ont été proposées dans la littérature pour modéliser ce signal d'excitation en vue de sa modélisation par HMM. La plus simple est la transformation de ce signal en un train d'impulsion à la fréquence fondamentale pour les sons voisés et en un bruit blanc pour les sons non voisés. Ce modèle n'a donc qu'un seul paramètre qui est la

fréquence fondamentale (égal à 0 lors de la synthèse de sons non-voisé). Cependant, il aboutit à une voix de synthèse très "robotique" (activité glottique peu réaliste). Plusieurs autres approches ont donc été proposées pour améliorer ce modèle. On citera notamment les approches dites par excitation mixte (Yoshimura et al. 2001 et Maia et al. 2007), et l'approche DSM (Drugman, Wilfart et Dutoit 2009) qui décomposent le signal d'excitation en une composante périodique et une composante apériodique (bruit) (ces approches sont proches du modèle harmonique plus bruit décrit ci-après), les approches exploitant un dictionnaire de signaux d'excitation (Drugman et al. 2009, Raitio et al. 2011) et les approches basées sur une modélisation explicite de l'onde de débit glottique (Lanchantin, Degottex et Rodet 2010).

3.3.4 Modèle "harmonique plus bruit"

Dans ce travail, nous adoptons une autre approche pour le paramétrage du signal de parole. Il s'agit du modèle harmonique plus bruit, proposé par (Stylianou 1996), et mis en œuvre dans (Hueber 2009) pour la conversion de parole silencieuse par HMM.

Contrairement à ce qui a été présenté dans la section précédente, cette approche ne repose pas sur une décomposition explicite source-filtre du signal de parole. Il s'agit de modéliser une trame de signal (considérée stationnaire) comme la somme d'un signal sinusoïdal $h(n)$ (composante périodique représentant la fréquence fondamentale et ses harmoniques) et d'un signal apériodique (on la qualifie de composante de "bruit") $b(n)$ décrivant les phénomènes apériodiques de la parole (comme par exemple les bruits de friction dans la production des fricatives), tels que $s(n) = h(n) + b(n)$. Ce modèle est donc appelé le modèle harmonique plus bruit, ou HNM (pour *Harmonic plus Noise Model* en anglais).

Nous rappelons maintenant les principes théoriques de ce modèle, tels que proposés dans (Stylianou 1996). La première étape est l'estimation du voisement, et pour les sons voisés de la fréquence fondamentale f_0 du signal. Plusieurs techniques peuvent être envisagées, telles que le *zero crossing rate* pour la détection du voisement ou l'autocorrélation pour la détection de la fréquence fondamentale (ou des techniques plus évoluées comme YIN (De Cheveigné et Kawahara 2002)). Pour les sons voisés, le spectre est ensuite décomposé en une partie harmonique, entre 0Hz et f_{mv}, la fréquence de voisement maximale et une partie bruitée, entre f_{mv} et $\frac{f_e}{2}$ (f_e étant la fréquence d'échantillonnage du signal). On note L le nombre d'harmoniques de f_0 incluses entre $f = 0Hz$ et $f = f_{mv}$ et $\{A_k\}_{k\in[-L,L]}$ leur amplitude complexe. La partie harmonique est modélisée par un modèle sinusoïdal ($\lfloor x \rfloor$ correspond à la

fonction partie entière de x) :

$$\begin{aligned} h(n) &= \sum_{k=-L}^{L} A_k(n_a^i) e^{j2\pi k(n-n_a^i)f_0(n_a^i)} \\ \text{avec} \quad n_a^{i+1} &= n_a^i + 1/f_0(n_a^i) \\ L &= \lfloor f_{mv}(n_a^i)/f_0(n_a^i) \rfloor \\ A_k &\in \mathbb{C} \text{ et } A_{-k} = A_k^* \end{aligned} \tag{3.2}$$

L'analyse des paramètres du modèle harmonique se fait de façon *pitch-synchrone* (i.e. deux instants d'analyse consécutifs n_a^i et n_a^{i+1} sont séparés d'une période fondamentale). Durant une période fondamentale, les paramètres (f_0, f_{mv}, A_k, l'amplitude des harmoniques) du modèle ne changent pas. On note $f_0(n_a^i)$ et $f_{mv}(n_a^i)$ la fréquence fondamentale et la fréquence maximale de voisement à $n = n_a^i$. Les amplitudes complexes $\{A_k\}_{k\in[-L,L]}$ sont calculées sur une fenêtre de pondération ω centrée sur n_a^i et d'une longueur $2N$ avec $N = \dfrac{1}{f_0(n_a^i)}$ selon la formule 3.3. Les différents paramètres du modèle sont représentés sur la Figure 3.5.

$$A_k = \frac{\sum_{n=n_a^i-N}^{n_a^i+N} \omega^2(n) s(n) e^{-j2\pi k f_0 n}}{\sum_{n=n_a^i-N}^{n_a^i+N} \omega^2(n)} \tag{3.3}$$

La partie bruit b d'une trame de signal voisée correspond à l'information qui n'a pas été capturée dans la partie harmonique h. Elle correspond donc au résidu du signal s après la modélisation sinusoïdale :

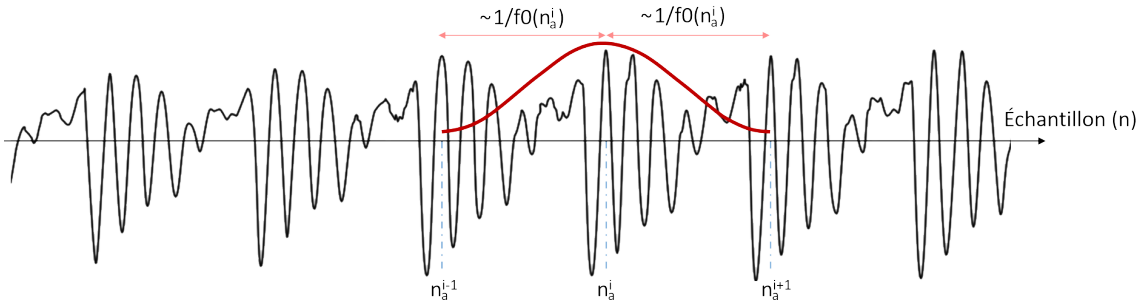
$$b(n) = s(n) - h(n) \tag{3.4}$$

Pour utiliser le modèle HNM dans le cadre de la synthèse par HMM, (Hueber 2009) [1] modélise les parties harmoniques et bruit par des modèles auto-régressifs (LPC), pour les sons voisés (pour les sons non voisés, un seul modèle LPC est utilisé). Cela permet d'obtenir un nombre constant de paramètres quelque soit la structure harmonique du spectre et la fréquence maximale de voisement.

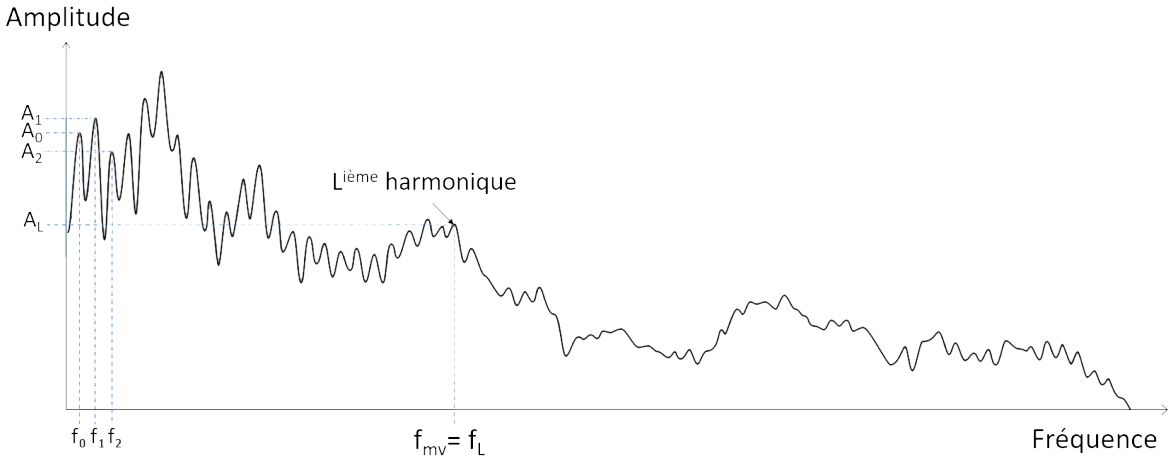
Un filtre auto-régressif d'ordre p est un filtre tout-pôle de la forme suivante :

$$F(z) = \frac{G}{A(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.5}$$

1. Cette approche s'appuie sur les travaux réalisés dans le cadre du projet SYMPATEX : `http://perso.telecom-paristech.fr/chollet/Projets/SYMPATEX/res_d76_ap99.htm`



(a) Évolution de l'amplitude d'un signal de parole durant une voyelle (a) en fonction du temps. Sur la figure, sont indiqués les instants n_a^i d'analyse, la fenêtre de *Hamming* ω pour l'analyse des coefficients du modèle sinusoïdal de taille $2 \times \dfrac{1}{f_0(n_a^i)}$



(b) Spectre de puissance associé à la Figure 3.5a. Sur cette figure sont représentés les amplitudes A_k des harmoniques, le L^{ieme} harmonique, la fréquence fondamentale f_0 et la fréquence maximale de voisement f_{mv}

FIGURE 3.5 – Évolution temporelle et spectre de puissance d'une voyelle [a]. Sur ces figures sont représentés les différents paramètres intervenant dans la modélisation HNM du signal.

où G est le gain du filtre. Les paramètres de ce modèle sont ensuite représentés dans l'espace des LSP (*Line Spectral Pairs*). Cet espace est préféré à l'espace des paramètres LPC car une altération d'un des coefficients (lors par exemple de l'inférence des paramètres par HMM) introduit une modification "locale" sur le spectre LSP, tandis qu'elle est potentiellement "globale" sur le spectre LPC (qui, de plus, peut devenir instable). Les coefficients LSP s'obtiennent à partir des coefficients LPC en calculant les racines des polynômes P et Q définis de la façon suivante :

$$\begin{aligned} P(z) &= A(z) + z^{p+1}A(z^{-1}) \\ Q(z) &= A(z) - z^{-(p+1)}A(z^{-1}) \\ \text{avec} \quad & A(z) \text{ et } p \text{ définis à l'Équation 3.5} \end{aligned} \tag{3.6}$$

Les racines de P et de Q ont les propriétés suivantes :

- Elles se situent sur le cercle unité (et sont donc de la forme $e^{j\omega_i}$).
- Les racines de P et celles de Q sont entrelacées sur le cercle unité.
- Les racines peuvent être regroupées par paires de complexes conjugués.
- Étant sur le cercle unité et par paires de complexes conjugués, les LSP peuvent s'exprimer sous la forme d'une pulsation ω_i comprise dans l'intervalle $[0; \pi]$. On parle alors de LSF (*Line Spectral Frequencies*).

Dans le cas de la modélisation HNM, une observation acoustique est donc composée de paramètres suivants :

- le gain G_h et les coefficients $LSF_h = [lsf_{h_0}, ..., lsf_{h_{N_h}}]$ de la partie harmonique.
- le gain G_b et les coefficients $LSF_b = [lsf_{b_0}, ..., lsf_{b_{N_b}}]$ de la partie bruit.
- la fréquence fondamentale f_0.

En phase de synthèse, la composante harmonique d'une trame de synthèse est obtenue par filtrage d'un train d'impulsions espacées de $1/f_0$, par un filtre AR dérivé des coefficients LSF_h. La composante bruit est obtenue par filtre d'une trame de bruit blanc par un filtre AR dérivé des coefficients LSF_b. La trame du signal audio de synthèse est obtenue par sommation des composantes harmonique et bruit, dans le domaine temporel. La Figure 3.6 résume les phases d'analyse et de synthèse HNM.

3.3.5 Modèle de Markov Caché

Un HMM est un modèle statistique qui peut être décrit comme une machine à états finis. Il s'agit d'un modèle génératif qui change d'état à chaque instant t selon une certaine probabilité dite de transition et émet à ce moment-là une observation $\mathbf{o_t}$. Cette dernière peut être vue comme la réalisation d'une variable aléatoire suivant une densité de probabilité dite d'émission. Un HMM à N états $\{q_i\}_{i\in[1,N]}$ est donc caractérisé par un ensemble de paramètres $\lambda = (A, B, \Pi)$ avec :

- $A = \{a_{i,j}\}_{\substack{i\in[1,N] \\ j\in[1,N]}}$ l'ensemble des probabilités de transition où $a_{i,j}$ est la probabilité de passer de l'état i à l'état j.

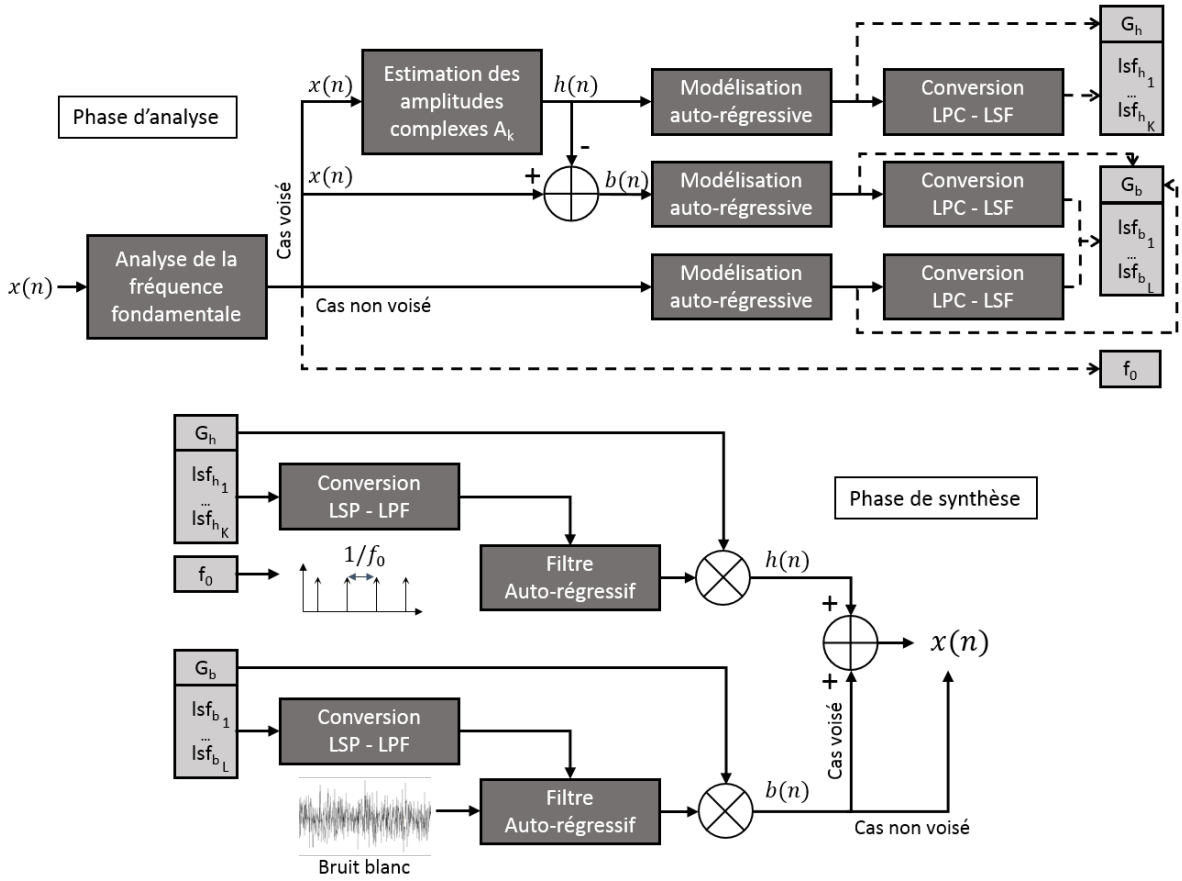


FIGURE 3.6 – Schéma bloc représentant le fonctionnement de l'analyse et de la synthèse d'un signal de parole dans le cadre de la modélisation "harmonique plus bruit".G_b et G_h sont respectivement les gains des modèles auto-régressifs des parties bruit et harmonique. L et K sont le nombre de coefficients LSF permettant de modéliser respectivement la partie bruit et la partie harmonique. Figure extraite de (HUEBER 2009).

— $B = \{b_i(\mathbf{o}_t)\}_{i\in[1,N]}$ l'ensemble des densités de probabilités d'émission pour chaque état q_i
— $\Pi = \{\pi_i\}_{i\in[1,N]}$, l'ensemble des probabilités a priori, pour chaque état i

Dans le cas de la modélisation d'une séquence temporelle, chaque état décrit par exemple une partie caractéristique de la séquence (par exemple pour une note piano : l'attaque, le corps et la chute). Une représentation graphique d'un HMM à 5 états avec une topologie "gauche-droite", et sans saut d'état, est fourni à la Figure 3.7. Il s'agit notamment de la topologie qui sera utilisée pour la modélisation des séquences d'observations acoustiques pour chaque classe de phonème.

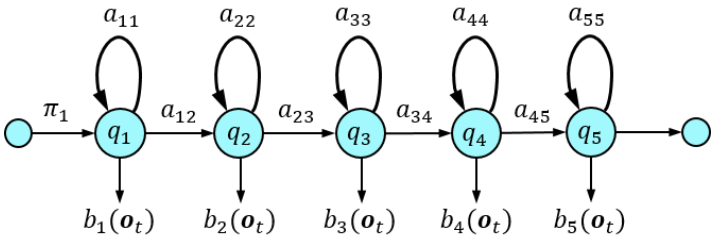


FIGURE 3.7 – Représentation schématique d'un HMM à 5 états émetteurs, avec un état initial et un étal final non émetteurs de trames, topologie "gauche-droite", sans saut d'états (topologie utilisée classiquement dans le cadre de la synthèse vocale par HMM). Cette topologie impose une durée d'au moins 5 trames.

Dans le cadre de la synthèse par HMM, une forme classique pour les densités de probabilités d'émission par état $\{b_i(\mathbf{o}_t)\}$ est une loi normale multivariée définie telle que :

$$\begin{aligned} b_i(\mathbf{o}_t) &= \mathcal{N}(\mathbf{o}_t; \mu_i, \mathbf{\Sigma}_i) \\ &= \frac{1}{\sqrt{(2\pi)^2|\mathbf{\Sigma}_i|}} \exp\{-\tfrac{1}{2}(\mathbf{o}_t - \mu_i)^\intercal \mathbf{\Sigma}_i^{-1}(\mathbf{o}_t - \mu_i)\} \end{aligned} \tag{3.7}$$

avec, en notant d la dimension du vecteur d'observation (acoustique), μ_i un vecteur de moyennes de taille d et $\mathbf{\Sigma}_i$ une matrice de covariance de taille $d \times d$ (en synthèse, on utilisera typiquement des matrices de covariances diagonales afin de limiter le nombre de paramètres du modèle). En modélisation par HMM, on présuppose que chaque séquence d'observations du corpus d'apprentissage a été "générée" par un HMM. On parle alors de modèle de Markov "caché", car la séquence d'états ayant été suivie pour générer la séquence d'observation n'est pas directement accessible à partir de cette dernière. Il s'agit d'une variable latente dont une estimation sera nécessaire pour l'apprentissage des paramètres du modèle HMM. Ce dernier s'effectue classiquement à l'aide de l'algorithme *Baum-Welch*, qui est une forme particulière de l'algorithme *Expectation-Maximization* pour les HMM. Cet algorithme n'est pas détaillé ici, mais une description complète peut être trouvée dans (BILMES 1998).

Dans la section suivante, nous décrivons la mise en œuvre des HMM pour la modélisation des trajectoires d'observations acoustiques, dans le cas de la synthèse vocale.

3.3.6 Modélisation HMM pour la synthèse vocale

3.3.6.1 Étiquetage contextuel des phonèmes

Dans le cadre de la synthèse paramétrique statistique, un HMM modélise l'évolution des observations acoustiques pour chaque classe phonétique, et ce, pour un contexte linguistique bien précis. En effet, on modélisera par exemple les trajectoires de paramètres acoustiques pour le phonème [a] lorsqu'il est précédé d'un [i] et suivi un [t], et en fonction de sa position dans la syllabe, dans le mot, voire dans la phrase. On distinguera d'une part les informations contextuelles dites segmentales, et d'autre part les informations contextuelles dites suprasegmentales. Les premières portent sur le contexte phonétique adjacent (typiquement les 2 phonèmes précédents et les 2 phonèmes suivants, également appelé quinphone). Elles jouent principalement un rôle pour la génération de la structure formantique, tel que montré par (Le Maguer 2013) pour le français, en prenant notamment en compte les phénomènes de coarticulation. Les secondes (suprasegmentales) intègrent notamment des informations sur la structure syntaxique de la phrase à synthétiser (position de chaque mot, accentuation des syllabes, etc). Elles jouent donc un rôle primordial dans la génération du contenu prosodique, tel que mentionné par (Büring 2013) et (Tsai et al. 2014).

À titre d'exemple, le Tableau 3.1 renseigne sur les descripteurs contextuels utilisés pour la synthèse du français dans le cadre de cette thèse. Ils sont classés en fonction du niveau auquel ils se rapportent (phonème, syllabe, mot, phrase). Pour indiquer la position d'une unité linguistique (e.g. un phonème) dans une unité de taille plus grande (e.g. un mot), on fait intervenir les notions de "comptage progressif" pour le comptage de la gauche vers la droite, et de "comptage rétrogressif" pour le comptage de la droite vers la gauche. Par exemple, si on considère la position du phonème /i/ dans le mot guitare (/gitaʁ/), le comptage progressif donne 2 et le comptage rétrogressif donne 4.

Notons dès maintenant que certains descripteurs contextuels peuvent porter sur le contexte droit du mot courant, contexte a priori non accessible en synthèse incrémentale. Ces descripteurs sont marqués en gras dans le Tableau 3.1 (pour certains d'entre eux, les phonèmes suivants par exemple, la connaissance du mot suivant n'est nécessaire que lorsque le phonème à synthétiser est proche de la fin du mot). La gestion en synthèse incrémentale de l'absence potentielle de descripteurs sur le contexte droit fera l'objet de la Section 3.4.2.

Dans la suite de ce document, nous nommerons *label* un phonème et ses descripteurs contextuels (segmentaux et suprasegmentaux, en contexte gauche et droit). Dans le formalisme utilisé par le *toolkit* HTS, on notera par exemple :
`m%e^-m+i=s@1_3/A:2/B:3@1-2&2-20=i/C:3/D:Cco-1/E:Adq+2@2+13-:DC/F:Npr-1`
le label associé au phonème [m] dans le contexte suivant : "Mais **m**ister Fogg (...)"

- le phonème [m] est précédé des 2 phonèmes [m] et [ɛ], suivi des 2 phonèmes [i] et [s]. Le phonème [m] est en position 1 dans la syllabe courante (comptage progressif) et en position 3 en partant de la fin de la syllabe courante (comptage rétrogressif).
- La syllabe précédente (`/A:`...) contient 2 phonèmes.
- La syllabe courante (`/B:`...) contient 3 phonèmes, elle est en position 1 dans le mot

- Phonème (7 valeurs)
  - Phonème à synthétiser (phonème actuel) [41]
  - Identité des deux phonèmes précédents et des **deux phonèmes suivants** [41]
  - Position du phonème actuel dans la syllabe actuelle (comptage progressif et rétrogressif) [7]
- Syllabe (7 valeurs)
  - Nombre de phonèmes dans les syllabes actuelle précédente et **suivante** [7]
  - Identité de la voyelle dans la syllabe actuelle [15]
  - Position de la syllabe actuelle dans le mot actuel (comptage progressif et rétrogressif) [9]
  - Position de la syllabe actuelle dans la phrase (comptage progressif et **rétrogressif**) [30]
- Mot
  - Classe lexicale des mots actuel, précédent et **suivant** [25]
  - Nombre de syllabes des mots actuel précédent et **suivant** [9]
  - Position du mot actuel dans la phrase (comptage progressif et **rétrogressif**) [25]
- Phrase
  - **Type de phrase** (déclaration, exclamation, question introduite, question totale, continuation) [5]

TABLE 3.1 – Liste des descripteurs contextuels utilisés pour la synthèse du français dans le cadre de cette thèse. Le nombre de paramètres associé à un niveau segmental est précisé entre parenthèses. Le nombre de valeurs possibles pour chacun des paramètres de la ligne correspondante est indiqué entre crochets.

courant (comptage progressif) et en position 2 en partant de la fin du mot courant (comptage rétrogressif). La syllabe courante est en position 2 dans la phrase courant et en position 20 en partant de la fin de la phrase. La voyelle de la syllabe courante est un [i].
— La syllabe suivante (`/C:`...) contient 3 phonèmes.
— Le mot précédent (`/D:`...) est une conjonction de coordination contenant 1 syllabe.
— Le mot courant (`/E:`...) est un adjectif qualificatif composé de 2 syllabes. Il est en position 2 dans la phrase courante et en position 13 en partant de la fin de la phrase courante. La phrase courante est de type déclarative.
— Le mot suivant (`/F:`...) est un nom propre constitué de 1 syllabe.

3.3.6.2 Modélisation explicite de la durée des phonèmes

Un aspect important en synthèse vocale est la durée de chaque unité linguistique (syllables, phonèmes, etc.). Dans le cadre de la synthèse par HMM, la modélisation de la durée s'effectue au niveau du phonème. À chaque état de chaque HMM est associé un modèle dit "de durée". Il s'agit d'un modèle probabiliste qui décrit la dynamique de la séquence d'état que le HMM doit suivre pour générer une séquence d'observations. Ce modèle prend classiquement la forme d'une loi normale pour chaque état du HMM (Yoshimura et al. 1998). Dans cette approche, la probabilité a priori d'observer un certain état ne dépend alors plus uniquement de l'état précédent, mais d'un modèle de durée explicite. On ne parle alors plus de modèles Markoviens mais de modèles semi-Markoviens, que nous notons dans la suite HSMM (pour *Hidden Semi-Markov Model*). Les paramètres d'un HSMM sont donc, pour chaque état :
— les probabilités d'émission.
— la moyenne et la variance du modèle de durée (qui remplacent les probabilités de transition des modèles HMM).

L'estimation de ces paramètres s'effectue à l'aide de l'algorithme Baum-Welch modifié pour prendre en compte les modèles de durée. Une description complète de l'algorithme d'apprentissage d'un HSMM est fournie dans (Yu 2010).

3.3.6.3 Modélisation multi-flux

En synthèse paramétrique, plusieurs types de paramètres doivent être modélisés, tels que les paramètres décrivant le contenu spectral (e.g. coefficients MGC dans le cas de la décomposition mel-cepstrale) et les paramètres relatifs à la source (e.g. fréquence fondamentale). On parle alors de modélisation et (par analogie) de modèles HMM multi-flux. Cela se formalise au niveau de la loi de probabilité d'émission qui s'écrit alors : $b_i(\mathbf{o}_t) = \prod_s \mathcal{N}(\mathbf{o}_{t,s}; \mu_{i,s}, \mathbf{\Sigma}_{i,s})$ avec $\mathbf{o}_t = [\mathbf{o}_{t,s1} ... \mathbf{o}_{t,sS}]$ un vecteur d'observation dans lequel sont concaténés les différents flux de paramètres.

3.3.6.4 Modélisation de la fréquence fondamentale

Certains paramètres comme par exemple la fréquence fondamentale ou les paramètres relatifs à la composante harmonique du modèle HNM ne sont pas définis de façon continue (ils ne le sont que pour les sons voisés). La modélisation d'une trajectoire discontinue nécessite une autre adaptation des procédures d'apprentissage des HMM. L'approche la plus classique en synthèse par HMM est le *Multi-Space probability Distribution* (MSD) proposé par (Tokuda, Masuko et Miyazaki 2002). Cette approche, qui ne sera pas entièrement re-détaillée ici, décompose chaque densité de probabilité comme une combinaison d'une distribution discrète (pour modéliser le caractère voisé/non-voisé de chaque trame) et d'une distribution continue (pour modéliser les variations dans $\mathbb{R}^*$ de la fréquence fondamentale).

(Suni et al. 2013) proposent une autre approche pour modéliser l'évolution de la fréquence fondamentale dans le cadre de la synthèse par HMM. Cette approche, que nous avons également évaluée dans ce travail de thèse pour la synthèse incrémentale par HMM, s'appuie sur la transformée en ondelettes continues. Nous résumons à présent cette approche. Soit $x(t) \in L^2(\mathbb{R})$ une fonction réelle du temps d'énergie finie, sa transformée en ondelettes s'écrit :

$$\sigma_{x,\phi}(t,a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(u)\overline{\phi}(\frac{u-t}{a})du \tag{3.8}$$

avec ϕ l'ondelette mère (une fonction oscillante d'énergie finie et d'intégrale nulle) que l'on peut dilater ou compresser d'un facteur d'échelle $a \in \mathbb{R}$ (analogue à une fréquence). $\overline{\phi}(t)$ est le complexe conjugué de $\phi(t)$.

Si l'ondelette est compressée, $(a > 1)$, les variations rapides de $x(t)$ seront capturées, si l'ondelette est dilatée $(a < 1)$, les variations lentes de $x(t)$ seront capturées. En choisissant un jeu de n facteurs d'échelle, $x(t)$ peut être décomposé en n fonctions décrivant les variations plus ou moins rapides de $x(t)$. Plus de détails sur la transformation en ondelettes continues peuvent être trouvés dans (Mallat 1999).

La reconstruction du signal $x(t)$ à partir de sa transformée s'obtient de la façon suivante :

$$x(t) = \frac{1}{C_\Phi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sigma_{x,\phi}(u,c)\phi_c(t-u)\frac{dudc}{c^2} \tag{3.9}$$

avec $\Phi(\nu)$ la transformée de Fourier de l'ondelette mère $\phi(t)$ et $C_\Phi = \int_{-\infty}^{+\infty} |\Phi(\nu)|^2 \frac{d\nu}{\nu}$.

Lors de la reconstruction, en limitant le domaine sur lequel on intègre le facteur d'échelle, on peut séparer les différentes bandes de fréquence du signal (voir partie inférieure de la Figure 3.8).

Les discontinuités de la fréquence fondamentale dues au voisement pouvant se répercuter sur les hautes fréquences et sur les basses fréquence, (Suni et al. 2013) proposent d'interpoler le signal sur les régions non voisées.

La Figure 3.8 illustre la décomposition en différentes bandes de fréquence d'un signal de fréquence fondamentale. On peut voir, sur la partie supérieure de la figure, la courbe

de fréquence fondamentale à analyser, sur la partie centrale, le logarithme de la fréquence fondamentale interpolée sur les régions non voisées et centré. Enfin, la courbe de la partie inférieure de la Figure 3.8 représente quatre bandes de fréquence du signal.

(Ribeiro, Yamagishi et Clark 2015) ont mené une série de 4 expériences perceptives visant à comparer l'approche MSD et l'approche par ondelettes continues pour la modélisation de la fréquence fondamentale dans le cadre de la synthèse par HMM (qualité de mise en emphase de certains mots, mesure de similarité avec la parole naturelle, *ranking* de type MUSHRA et MOS). Les résultats montrent un avantage de l'approche par ondelettes. Aussi, c'est cette approche que nous utiliserons dans la suite de ce travail pour la mise en place de notre synthétiseur vocal incrémental.

3.3.6.5 Entraînement des modèles contextuels

L'entraînement des modèles HSMM contextuels pour la synthèse vocale suit une procédure proche de celle utilisée pour la reconnaissance automatique de la parole. Cette procédure est illustrée à la Figure 3.9. Elle consiste, dans un premier temps, à estimer un modèle HSMM pour chaque classe phonétique hors-contexte. Ces modèles sont appelés CI-HSMM (pour *Context-independant HSMM*). Cette estimation s'effectue à l'aide de l'algorithme Baum-Welch, tout d'abord en considérant des modèles isolés (estimation des paramètres d'un modèle de façon indépendante des autres modèles) puis en modèles dit "connectés" (estimation conjointe des paramètres des modèles associés aux phonèmes d'une même phrase). Les modèles CI-HSMM sont ensuite utilisés pour initialiser les modèles "en contexte" (appelé CD-HSMM pour *Context-Dependant HSMM*). Cependant, compte tenu du nombre de descripteurs contextuels considérés (voir Tableau 3.1), le nombre d'occurrences par contexte est très faible (on parle en anglais de *sparse dataset*). Aussi, l'estimation de modèles contextuels CD-HSMM ne peut se faire de façon robuste. On procède donc à une étape dite de "partage d'états" (*state-tying*, Young, Odell et Woodland 1994). Cette étape consiste à regrouper les contextes linguistiques qui sont susceptibles d'être associés à des observations acoustiques proches. Dans le cadre de la synthèse par HMM, ce regroupement s'effectue au niveau des états des CD-HSMM. Il s'appuie sur un ensemble d'arbres de décision binaire, pour lesquels, à chaque nœud, est associé une "question" sur la valeur d'un descripteur contextuel (par exemple : "le mot suivant est-il un nom ?"). L'arbre de décision est construit en conservant, pour chaque nœud, le critère de séparation qui permet de maximiser la vraisemblance du modèle séparé. Ainsi, tous les labels parvenant à une même feuille de l'arbre de décision partageront plusieurs propriétés linguistiques (par exemple : "le phonème courant est une voyelle", "le mot suivant est un nom", etc.) et seront dits "liés" : ils partageront les mêmes paramètres, estimés à partir de toutes les observations acoustiques associées à ces labels.

Un exemple d'un tel arbre est donné en Figure 3.10.

En synthèse par HMM, ce regroupement s'effectue au niveau des états des modèles. Un arbre de décision est construit pour chaque état, et chaque flux. Un arbre est également construit pour regrouper les modèles de durées des états. Pour chaque arbre, la question associée à chaque nœud est choisie parmi l'ensemble des questions possibles comme celle qui

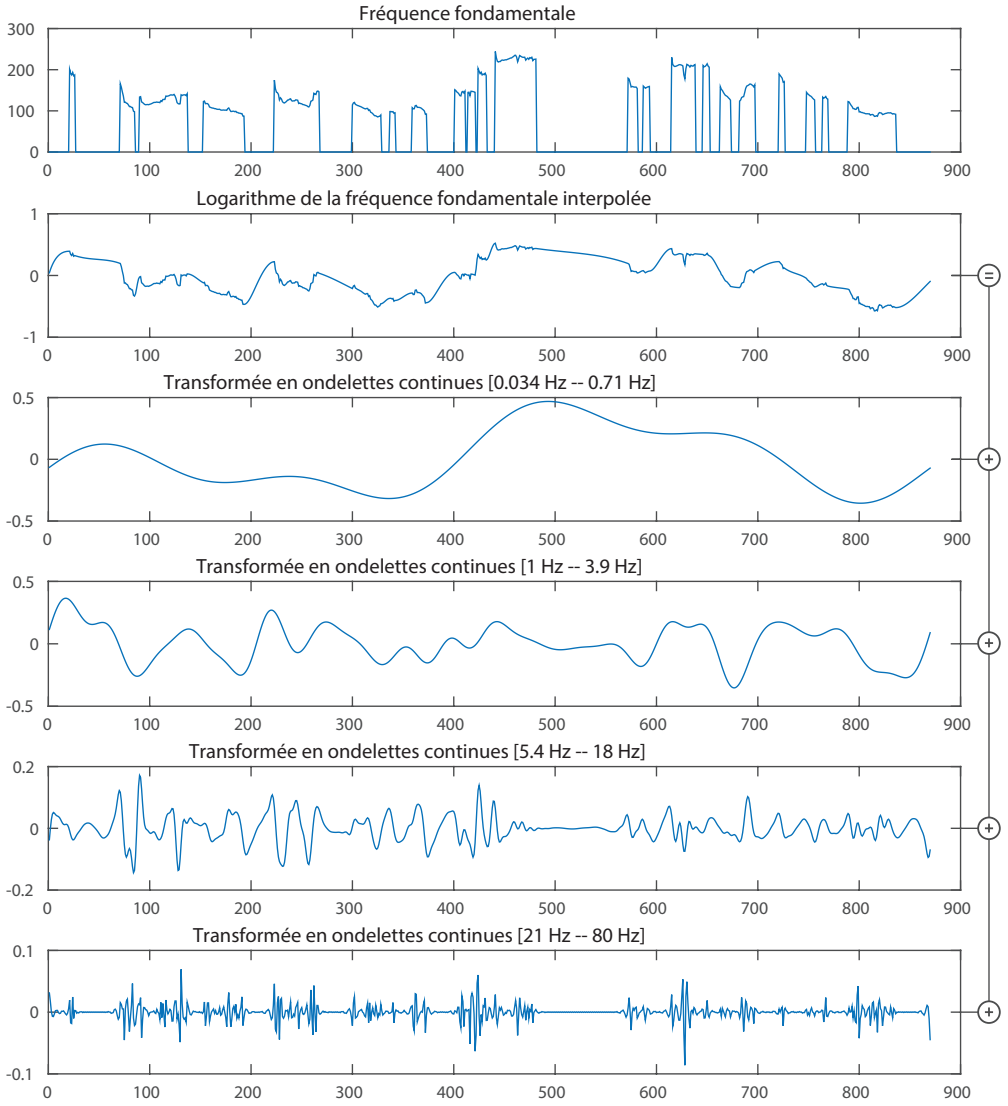


FIGURE 3.8 – Transformée en ondelettes continues d'un signal de fréquence fondamentale. La courbe supérieure représente l'évolution de la fréquence fondamentale du signal en fonction du temps, la courbe du second panneau représente le logarithme de la fréquence fondamentale interpolée sur les régions non voisées, centré et normé, et les courbes inférieures représentent les différentes échelles de la transformée en ondelettes continues. Les quatre échelles retenues décrivent respectivement, de haut en bas, les variations sur les plages : 0.03Hz – 0.7Hz ; 1Hz – 3.9Hz ; 5.4Hz – 18Hz ; 21Hz – 80Hz. L'erreur quadratique de reconstruction de $f_0(t)$ à partir de la transformation en ondelettes continues est de 0.9Hz

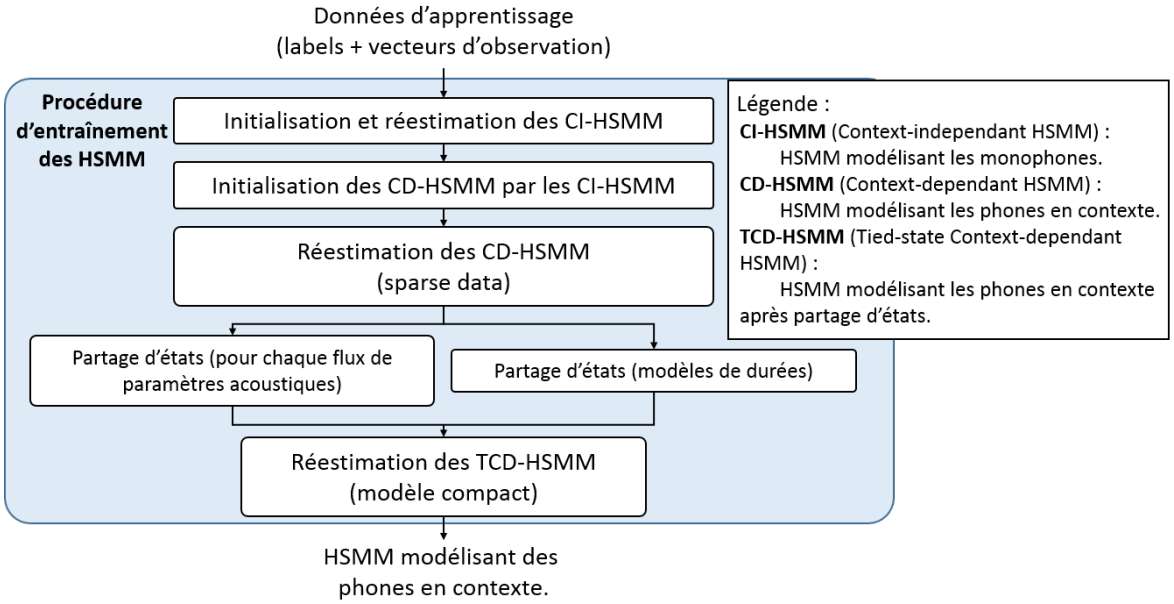


FIGURE 3.9 – Procédure d'entraînement des HSMM pour la synthèse de parole telle que proposée dans le toolkit HTS. On appelle monophone un phone seul (i.e. dépourvu de descripteurs contextuels). Les opérations de ré-estimation consistent à appliquer l'algorithme Baum-Welch pour réestimer les paramètres des HSMM. Les opérations de partage d'état (*state-tying* en anglais), sont détaillées ci-après.

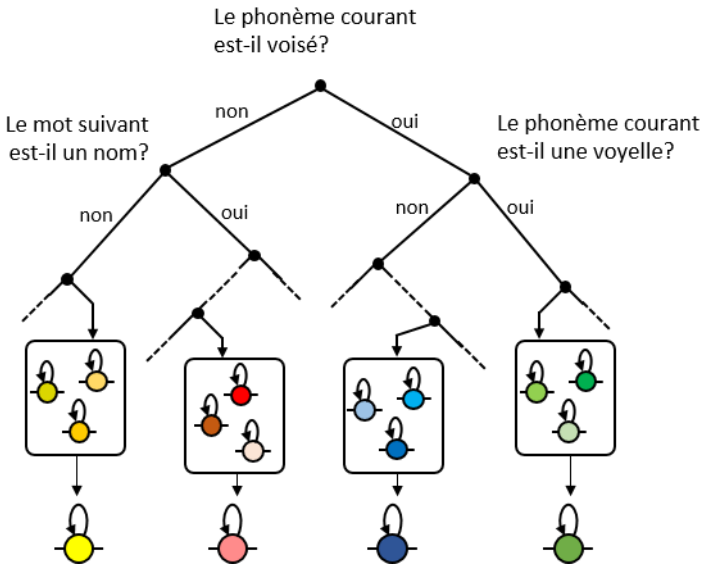


FIGURE 3.10 – Illustration du partitionnement de l'espace acoustique grâce à un arbre de décision

augmente la vraisemblance des modèles liés sur le corpus d'apprentissage. Ce processus de sélection de question (et donc de regroupement) est itéré pour construire successivement les différents niveaux de l'arbre. Dans le cadre du *toolkit* HTS, le critère *Minimum Description Length* (Shinoda et Watanabe 2000) est classiquement utilisé comme critère d'arrêt. Il permet d'obtenir un compromis entre le nombre de modèles liés (profondeur de l'arbre) et le nombre d'observations acoustiques utilisables pour l'estimation des paramètres de ces derniers.

Une fois le regroupement optimal déterminé pour chaque état et chaque flux, les paramètres des modèles liés sont ré-estimés à l'aide de l'algorithme Baum-Welch. À ce stade, on parle de modèle *Tied-state Context-Dependent* (TCD)-HSMM [2].

3.3.7 Synthèse

Dans cette section, nous décrivons l'inférence d'une séquence d'observations acoustiques, à partir d'une séquence de labels issue du module de TAL, dans le cadre de la synthèse par HMM (partie inférieure de la Figure 3.2).

3.3.7.1 Sélection des états du modèle HSMM de synthèse

Comme mentionné ci-avant, un label renseigne sur le phonème à synthétiser ainsi que sur son contexte segmental et suprasegmental. Étant donné le nombre de descripteurs contextuels considérés, il est très peu probable de disposer à l'issue de la phase d'apprentissage, d'un modèle HSMM pour chaque contexte à synthétiser. Aussi, un modèle HSMM associé à un label non vu dans le corpus d'apprentissage doit être construit. Cela s'effectue en exploitant les arbres de décisions binaires mis en place pendant la phase d'apprentissage pour le regroupement des modèles (*tree-based state tying*). Pour chacun des états du modèle HSMM à synthétiser, et pour chaque flux, on fait "parcourir" au label associé les différentes branches de chaque arbre de décision, et on récupère l'état associé à la feuille de la dernière branche parcourue. Cette procédure est répétée pour chaque label de la phrase à synthétiser. Cette procédure de construction d'un nouveau modèle HSMM par sélection d'états est illustrée à la Figure 3.11.

3.3.7.2 Génération de paramètres

L'inférence d'une séquence de paramètres acoustiques $\hat{\mathbf{o}}$ à partir de la séquence de labels $\mathbf{w} = [w_1, ..., w_L]$ s'effectue à l'aide de l'algorithme *Maximum Likelihood Parameter Generation* ou MLPG (Tokuda et al. 2000). Cette inférence s'effectue au sens du maximum de vraisemblance tel que :

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} P(\mathbf{o}|\mathbf{w}, \hat{\lambda})$$

2. Dans le *toolkit HTS*, la procédure de partage d'état est classiquement effectué deux fois. On parle de *state-tying*, *state untying*, et *state-retying*.

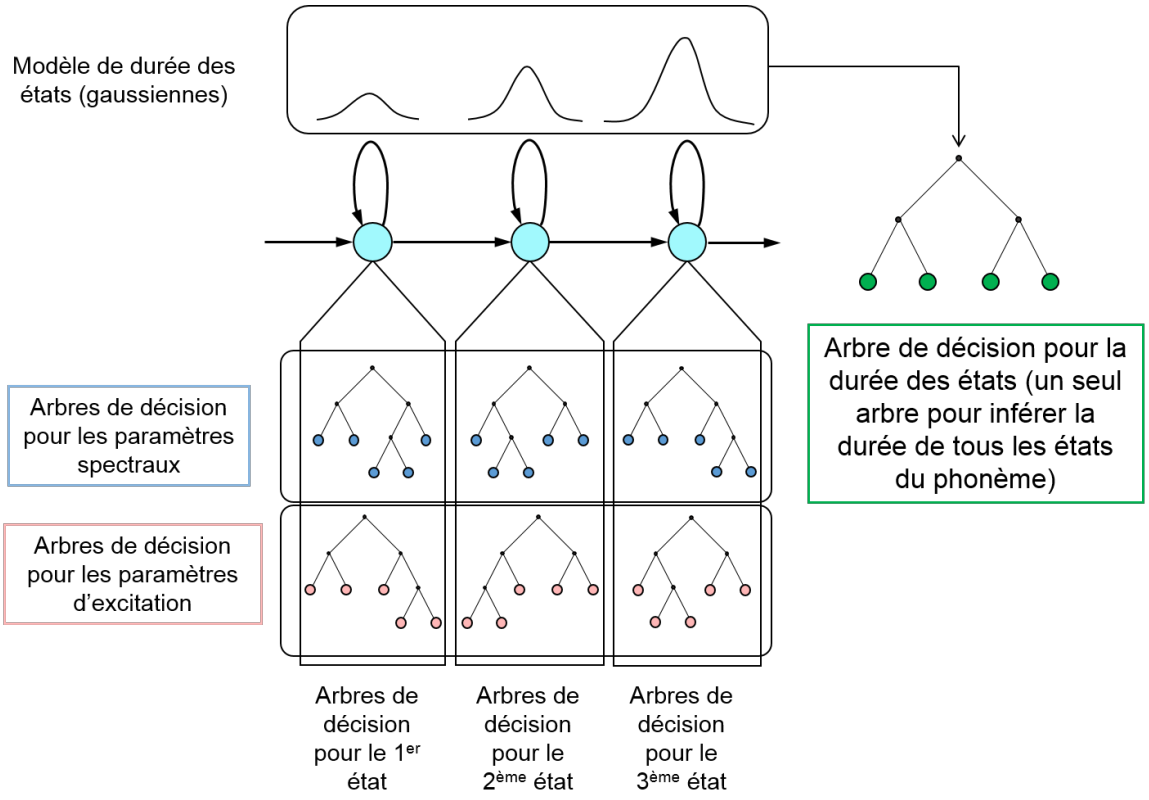


FIGURE 3.11 – Illustration du processus de construction d'un modèle HSMM associé à un contexte non vu dans le corpus d'apprentissage, par exploration des arbres de décisions et sélection d'états.

(avec $\hat{\lambda}$ les paramètres du modèle HSMM construit par concaténation des états sélectionnés à l'aide de la procédure décrite précédemment). En introduisant la séquence d'états $\mathbf{q} = [q_1, ..., q_N]$ qu'il faut "parcourir" pour générer N observations acoustiques, cette équation s'écrit :

$$\begin{aligned} \hat{\mathbf{o}} &= \underset{\mathbf{o}}{\text{argmax}}\ P(\mathbf{o}|w, \hat{\lambda}) \\ &= \underset{\mathbf{o}}{\text{argmax}} \sum_{\forall \mathbf{q}} P(\mathbf{o}, \mathbf{q}|w, \hat{\lambda}) \end{aligned} \tag{3.10}$$

Cette équation peut également s'écrire :

$$\begin{aligned} \hat{\mathbf{o}} &= \underset{\mathbf{o}}{\text{argmax}}\ P(\mathbf{o}|w, \hat{\lambda}) \\ &= \underset{\mathbf{o}}{\text{argmax}} \sum_{\forall \mathbf{q}} P(\mathbf{o}|w, \mathbf{q}, \hat{\lambda}) P(\mathbf{q}|w, \hat{\lambda}) \end{aligned} \tag{3.11}$$

En ne considérant que la séquence d'états la plus probable, on obtient :

$$\begin{aligned} \hat{\mathbf{o}} &= \underset{\mathbf{o}}{\text{argmax}}\ P(\mathbf{o}|w, \hat{\lambda}) \\ &= \underset{\mathbf{o}}{\text{argmax}}\ \underset{\mathbf{q}}{\text{argmax}}\ P(\mathbf{o}|w, \mathbf{q}, \hat{\lambda}) P(\mathbf{q}|w, \hat{\lambda}) \end{aligned} \tag{3.12}$$

Cette dernière peut se résoudre en 2 étapes successives :
— En déterminant $\hat{\mathbf{q}} = \underset{\mathbf{o}}{\text{argmax}}\ P(\mathbf{q}|w, \hat{\lambda})$
— En calculant $o \approx \underset{\mathbf{q}}{\text{argmax}}\ P(\mathbf{o}|\hat{\mathbf{q}}, \hat{\lambda})$

La première étape, c'est-à-dire la détermination de la séquence d'états se déduit des modèles de durées (voir Section 3.3.6.2), en déterminant la séquence d'états $\hat{\mathbf{q}}$ qui maximise $P(\mathbf{q}|w, \hat{\lambda})$. Dans le cas de modèles de durées gaussiens, on obtient $P(\mathbf{q}|w, \hat{\lambda}) = \prod_{i=1}^{N} \mathcal{N}(d_i; m_i, \sigma_i^2)$, avec d_i la durée associée à l'état q_i, donc une durée cible par état égale à m_i.

La seconde étape, c'est-à-dire l'inférence des trajectoires de paramètres acoustiques à partir de la séquence d'états estimée à la première étape, s'obtient à l'aide de l'algorithme *Maximum Likelihood Parameter Generation* (MLPG, Tokuda et al. 2000).

Cet algorithme peut se résumer comme suit. Tout d'abord, il présuppose qu'une observation acoustique $\mathbf{o}_t$ se décompose en une série de descripteurs (dits "statiques") $\mathbf{c}_t$ auxquels sont adjoints leur dérivées première et seconde, tel que :

$$\mathbf{o}_t = [\mathbf{c}_t^\intercal, \Delta\mathbf{c}_t^\intercal, \Delta^2\mathbf{c}_t^\intercal]^\intercal$$

avec classiquement :

$$\begin{aligned} \Delta\mathbf{c}_t &= 0.5(\mathbf{c}_{t+1} - \mathbf{c}_{t-1}) \\ \Delta^2\mathbf{c}_t &= \mathbf{c}_{t-1} - 2\mathbf{c}_t + \mathbf{c}_{t+1} \end{aligned} \tag{3.13}$$

La relation entre la séquence de descripteurs statiques et leurs dérivées peut s'écrire sous forme matricielle tel que :

$$\mathbf{O} = W \times \mathbf{C} \tag{3.14}$$

avec $\mathbf{C} = [c_1^\intercal, \ldots, c_N^\intercal]^\intercal$ et $\mathbf{O} = [o_1^\intercal, \ldots, o_N^\intercal]^\intercal$ des vecteurs colonnes de taille respectives 1xNd et 1x3Nd (avec d la dimension de c_t), et W une matrice dont la structure est explicitée à la Figure 3.12.

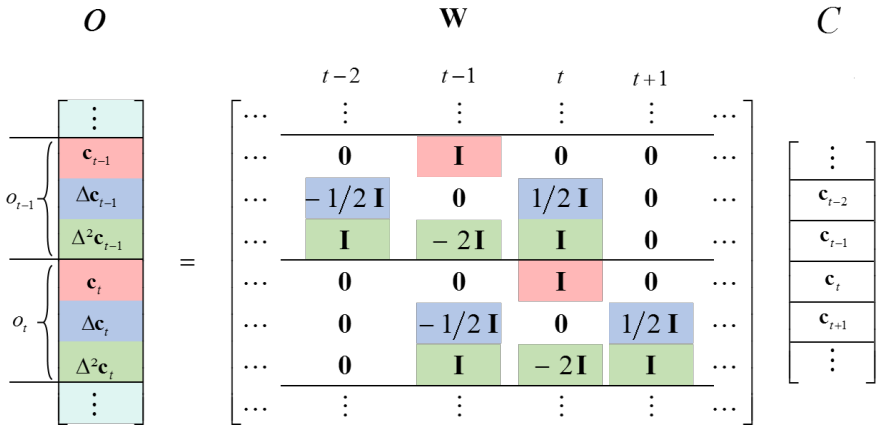


FIGURE 3.12 – Représentation matricielle de la relation entre la séquence d'observations (vecteurs statiques et dynamiques) et la séquence de vecteurs statiques. Sur cette figure, **I** représente la matrice identité de taille $d \times d$, avec d la dimension du vecteur de paramètres. **0** représente une matrice de 0 de taille $d \times d$. Figure inspirée de (HTS 2000

L'idée principale de l'algorithme MLPG est de rechercher la trajectoire de paramètres qui maximise la vraisemblance du modèle non seulement pour les descripteurs statiques mais également pour leurs dérivées première et seconde. Ceci garantit notamment l'absence de discontinuité, dans la séquence de paramètres inférée, lors du changement d'état. Formellement, cela s'obtient à l'aide de l'équation suivante :

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} P(\mathbf{W}\mathbf{c}|\hat{\mathbf{q}}, w, \hat{\lambda}) \tag{3.15}$$

En annulant la dérivée partielle par rapport à $\mathbf{c}$ du logarithme de 3.15, on obtient :

$$\begin{aligned} &\hat{\mathbf{c}} = (W^\intercal U^{-1} W)^\intercal W^\intercal U^{-1} M \\ \text{avec :} \quad &M = \left[\mu_{\mathbf{q1}}^\intercal, ..., \mu_{\mathbf{qT}}^\intercal\right]^\intercal \\ &U = diag\left[\Sigma_{\mathbf{q1}}, ..., \Sigma_{\mathbf{qT}}\right] \end{aligned} \tag{3.16}$$

Cette résolution permet d'obtenir une trajectoire continue telle que présentée en Figure 3.13.

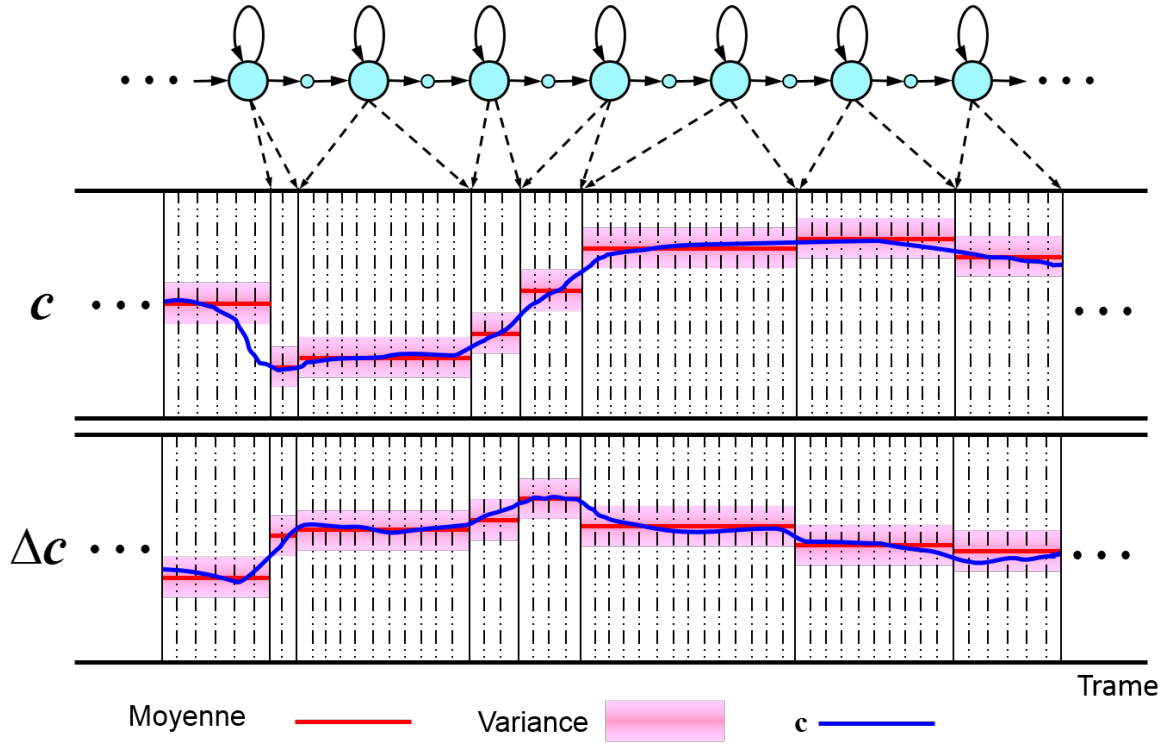


FIGURE 3.13 – Exemple de génération de paramètres par HMM à l'aide de l'algorithme MLPG. Figure extraite de (HTS 2000)

3.4 Synthèse incrémentale de la parole par HMM

3.4.1 État de l'art

Dans le paradigme de la synthèse incrémentale, la synthèse doit être déclenchée au fur et à mesure de la saisie du texte. Plusieurs adaptations sont nécessaires par rapport à une synthèse vocale non-incrémentale, dont l'unité linguistique élémentaire est la phrase. Tout d'abord, certains descripteurs contextuels se rapportent au contexte droit d'un mot. Une partie d'entre eux n'est donc, a priori, pas accessible en synthèse incrémentale. Ensuite, lors de la résolution de l'Équation 3.15, l'algorithme MLPG s'appuie sur l'ensemble des étiquettes phonétiques de la phrase (c'est-à-dire w) pour inférer les trajectoires de paramètres acoustiques. En synthèse incrémentale, la suite des étiquettes phonétiques se dévoilant progressivement, cet algorithme doit donc être adapté. Ces deux aspects ont déjà fait l'objet de différents travaux que nous résumons maintenant.

3.4.1.1 Travaux de Le Maguer et al.

Dans le cadre de la synthèse par HMM du français, (Le Maguer 2013) étudie l'impact des différents descripteurs contextuels sur la qualité segmentale et suprasegmentale. Il constate que l'estimation des paramètres spectraux (i.e. qualité segmentale) ne nécessite que la connaissance du "quinphone" (i.e. le phonème à synthétiser, les deux phonèmes précédents et les deux phonèmes suivants). L'estimation de la durée des phonèmes et de la fréquence fondamentale (qualité suprasegmentale) nécessite des connaissances à plus long terme comme par exemple, la position de la syllabe courante dans le syntagme.

Ces travaux étant réalisés dans le cadre de la synthèse TTS "classique" (c'est-à-dire non incrémentale), les descripteurs contextuels , même limités, sont toujours considérés comme connus lors de la synthèse. Dans ce travail de thèse, nous chercherons donc à vérifier les résultats de (Le Maguer 2013) dans le cadre de la synthèse incrémentale (pour laquelle un descripteur portant sur le contexte droit peut être manquant).

3.4.1.2 Travaux de Astrinaki et al.

Pour permettre la synthèse incrémentale, (Astrinaki 2014) propose de supprimer l'ensemble des descripteurs contextuels qui se rapportent au contexte droit. Les évaluations objectives et subjectives menées sur ce jeu réduit de descripteurs (portant donc uniquement sur le contexte gauche) indiquent une dégradation segmentale et suprasegmentale mais ne pénalisant que légèrement la qualité d'écoute par rapport à une synthèse utilisant un jeu de descripteurs complet.

Par ailleurs, (Astrinaki 2014) propose une résolution de l'algorithme MLPG sur une courte fenêtre de labels plutôt que sur l'ensemble de la phrase. Cette implémentation de l'algorithme MLPG à court terme permet d'obtenir des résultats comparables à l'implémentation originale proposée par (Tokuda et al. 2000).

3.4.1.3 Travaux de Baumann et al.

(Baumann 2014) propose un prototype complet de synthétiseur incrémental basé sur le système de dialogue incrémental InPro (Baumann et Schlangen 2012c). La stratégie qu'il propose pour synthétiser de la parole avec des descripteurs contextuels manquants consiste à les substituer par des valeurs par défaut. Ces dernières sont estimées sur le corpus d'apprentissage. Cette stratégie (que nous nommerons stratégie "Par Défaut") servira de stratégie de référence dans la suite de ce travail. Nous la détaillons maintenant.

Comme détaillé en Section 3.3.6.1, dans la plupart système TTS par HMM, l'unité élémentaire de parole modélisée est le phonème, accompagné de son contexte. Comme indiqué dans le Tableau 3.1, ce contexte est classiquement décrit par un jeu de descripteurs contextuels segmentaux (ex : phonème actuel, phonèmes précédents et suivants) et suprasegmentaux (ex : la classe lexicale du mot auquel appartient le phone à synthétiser, les classes lexicales des mots adjacents, la position du mot actuel dans la phrase).

Comme nous avons pu le voir en Section 3.3.6.5, il est quasiment impossible de construire un corpus d'apprentissage présentant suffisamment d'occurrences de chacun des contextes possibles. Une étape de partitionnement de l'espace acoustique, reposant sur l'utilisation d'arbres de décision, permet de rassembler des états acoustiquement proches pour estimer de façon robuste les paramètres des modèles HMM contextuels. Étant donné un contexte cible lors de la synthèse, ces mêmes arbres de décision sont exploités pour sélectionner la suite d'états HMM à utiliser pour l'inférence des trajectoires de paramètres acoustiques.

(Baumann 2014) propose d'utiliser ces arbres de décision pour calculer des valeurs par défaut qui seront utilisées lorsque certains descripteurs contextuels sont manquants (en synthèse incrémentale). La procédure peut être résumée de la manière suivante. Elle s'appuie tout d'abord sur une voix de synthèse entraînée à l'aide de la procédure standard décrite en Section 3.3.6.5.

Lors de l'apprentissage, l'ensemble des descripteurs contextuels (gauches et droits), sont supposés connus. Une "valeur par défaut" est ensuite attribuée à chaque descripteur contextuel renseignant sur le contexte droit, pouvant potentiellement être inconnu lors du traitement incrémental (descripteurs en caractères gras dans le Tableau 3.1). Cette "valeur par défaut" est calculée à partir du corpus d'entraînement en prenant tous les nœuds de l'arbre dont la question porte sur le contexte droit et en considérant les valeurs de tous les labels interrogés pour ce nœud. Pour les paramètres numériques (le nombre de syllabes avant la fin de la phrase, par exemple), la valeur par défaut correspondante est la moyenne numérique. Pour les descripteurs symboliques (la classe lexicale du mot suivant par exemple), la valeur par défaut retenue est la valeur la plus fréquente.

Cette stratégie permet donc d'utiliser dans un contexte incrémentale une voix de synthèse entraînée de façon classique et donc d'utiliser des voix déjà disponibles (par exemple les voix des systèmes HTS ou openMary, Schröder et Trouvain 2003). Cependant, cette stratégie ne permet pas d'exploiter l'ensemble des modèles disponibles d'une voix. En effet, en imposant une "valeur par défaut", certaines branches de l'arbre de décision ne sont plus du tout explorées

lors de la sélection des états pour l'inférence des trajectoires de paramètres acoustiques. Ce problème est illustré à la Figure 3.14. La finesse de la modélisation du corpus d'entraînement (s'appuyant sur l'ensemble des descripteurs contextuels) n'est donc pas entièrement exploitée lorsqu'elle est utilisée lors de la synthèse incrémentale.

Pour pallier cette limitation, nous proposons une autre stratégie pour la synthèse vocale incrémentale par HMM. Cette stratégie fait l'objet de la section suivante.

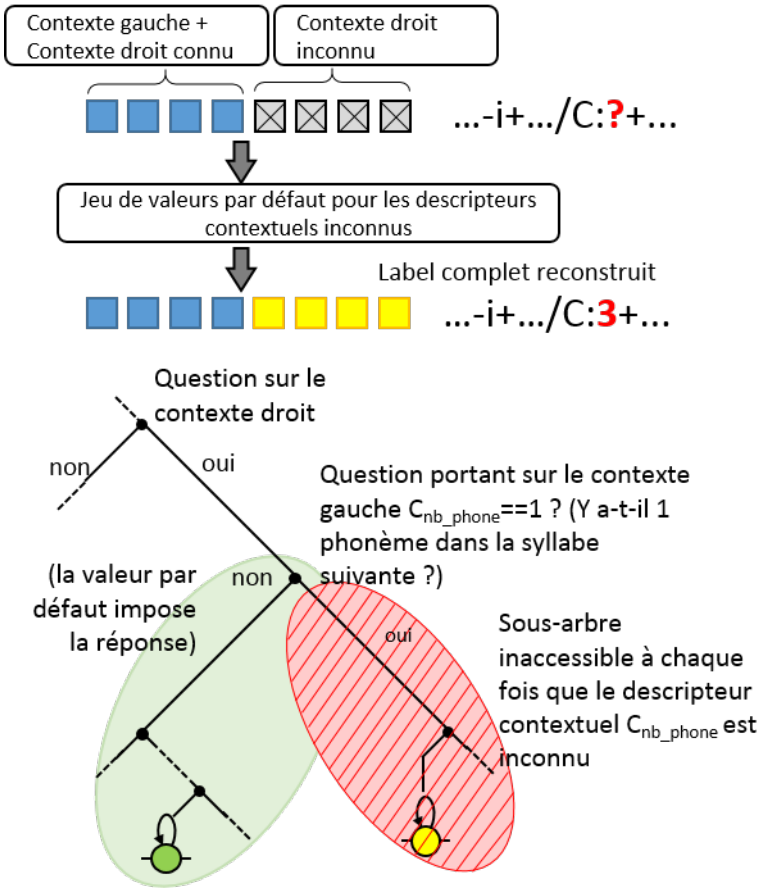


FIGURE 3.14 – Illustration des effets de la stratégie proposée par Baumann et al. ("Par Défaut") sur les arbres de décision. Sur la partie supérieure, les carrés bleus représentent les descripteurs contextuels et les gris les descripteurs manquants lors de la synthèse incrémentale. Les carrés jaunes sont les valeurs par défaut pour chacun des descripteurs contextuels inconnus. Le label suit le formalisme proposé dans HTS avec /i/ le phonème à synthétiser et /C :(...) les descripteurs contextuels se rapportant à la syllabe suivante.

3.4.2 Méthode proposée pour la synthèse incrémentale de la parole par HMM : stratégie "Joker"

Dans la suite de ce chapitre, nous décrivons la méthode que nous proposons pour la synthèse vocale incrémentale par HMM. Contrairement à la stratégie de Baumann et al., qui estime une

valeur par défaut pour les descripteurs contextuels manquants, nous proposons d'intégrer cette absence d'information sur le contexte droit au moment de la création de la voix de synthèse (c'est-à-dire au moment de l'estimation des paramètres des HMM).

3.4.2.1 Principe

Nous cherchons ici à créer des modèles spécifiques pour les contextes incomplets, c'est-à-dire des contextes pour lesquels un ou plusieurs descripteurs relatifs au contexte droit sont manquants. Cette nouvelle approche est nommée ici la "stratégie *joker*". Tout descripteur contextuel potentiellement manquant en synthèse incrémentale se voit attribuer une valeur "*joker*", indiquant explicitement qu'il est indéterminé. Lors de la phase d'apprentissage, et en particulier lors du partitionnement de l'espace acoustique (*tree-based state tying*), cette approche permet de regrouper des modèles qui partagent une incertitude sur un ou plusieurs descripteurs contextuels. On peut par exemple s'attendre à ce qu'un tel entraînement produise une prosodie neutre, dans des situations où la connaissance du contexte droit entraînerait une prosodie plus marquée.

L'avantage a priori de cette approche par rapport à la stratégie de Bauman et al. (décrite précédemment) est de présenter une cohérence entre les phases d'apprentissage et de synthèse. En effet, une incertitude sur la valeur d'un des descripteurs contextuels droits est ici traitée de la même manière lors des deux phases. En synthèse incrémentale, la stratégie proposée recrutera un modèle "moyen" en cas de descripteurs contextuels manquants alors que la stratégie de Baumann et al. choisira un modèle précis mais potentiellement mal adapté. C'est le cas lorsque la valeur par défaut, attribuée à un descripteur manquant lors de la synthèse, apparaît comme erronée une fois la phrase complètement saisie.

3.4.2.2 Implémentation

La stratégie proposée "Joker" s'implémente facilement dans le cadre d'un système de synthèse par HMM comme HTS. Dans un premier temps, le corpus d'entraînement est étiqueté en utilisant une valeur "Joker" (représentée dans notre implémentation sous la forme du symbole ##) pour chaque information contextuelle qui ne pourra pas être déterminée de façon univoque en considérant un traitement incrémental du texte. Ce traitement concerne principalement les informations contextuelles portant sur le mot suivant. À titre d'exemple, considérons l'information contextuelle "Nombre de phonèmes dans la syllabe suivante". Si l'on cherche à construire le label associé au dernier phonème d'un mot, l'information "Nombre de phonèmes dans la syllabe suivante" étant inconnue dans un contexte de synthèse incrémentale, la valeur ## lui est attribuée. Les modèles contextuels sont ensuite entraînés en utilisant la procédure décrite en 3.3.6.5 (de façon similaire à un entraînement de voix non-incrémentale), la valeur "joker" étant considérée comme une valeur possible pour les paramètres contextuels. Lors de la procédure de regroupement d'états basée sur des arbres de décision, nous proposons d'introduire une question spécifique afin de savoir si un paramètre est connu ou non. L'ajout d'une telle question permet de s'attendre à ce que des modèles ou des états qui partagent une

incertitude sur leurs paramètres soient regroupés pour entraîner des modèles présentant une incertitude sur un (ou plusieurs) paramètre(s) portant sur le futur. Le reste de la procédure d'entraînement des modèles HMM est semblable à l'entraînement classique de modèles HMM pour la synthèse de parole non-incrémentale. Comme présenté sur la Figure 3.15, l'un des principaux avantages de la stratégie proposée par rapport à la stratégie de Baumann et al. est qu'elle permet une meilleure utilisation des arbres de décision lors de la synthèse : si une question concernant une information "manquante" est posée au cours de la descente de l'arbre, les deux sous-arbres qui en résultent restent potentiellement accessibles : contrairement à la stratégie "Par Défaut", toutes les feuilles de l'arbre (états de modèles HMM) sont atteignables. Cela permet d'utiliser l'ensemble des modèles de phonèmes en contexte.

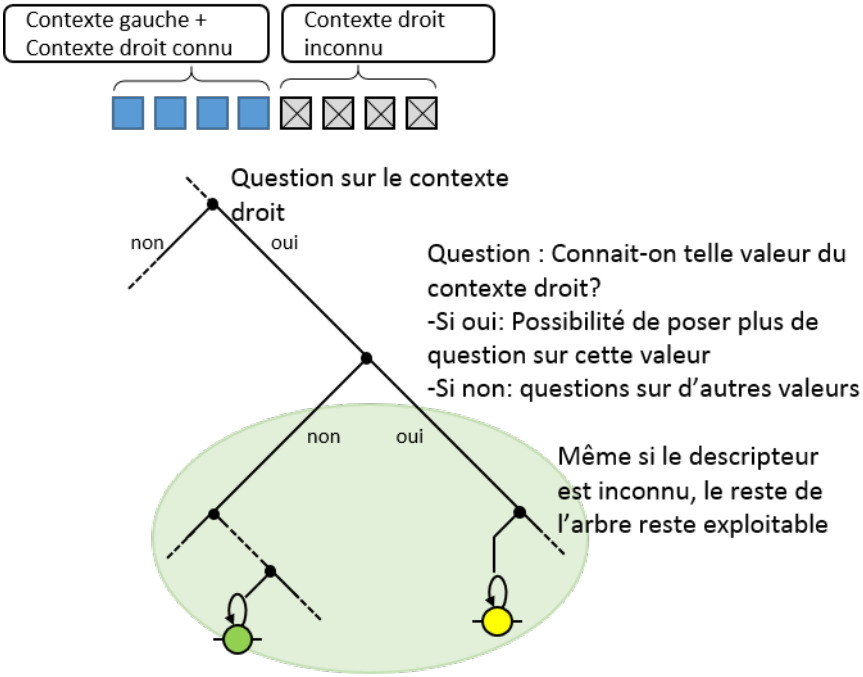


FIGURE 3.15 – Illustration des effets de la stratégie proposée pour la synthèse incrémentale sur l'exploitation des arbres de décision. Sur la partie supérieure, les carrés bleus représentent les descripteurs contextuels et les gris et les descripteurs inconnus. Les descripteurs inconnus sont utilisés en tant que "joker" lors du parcours de l'arbre de partitionnement.

3.5 Évaluation expérimentale

3.5.1 Mise en œuvre d'un système de synthèse par HMM pour le français

Pour évaluer la stratégie proposée pour la synthèse par HMM incrémentale, nous avons tout d'abord entraîné une voix de synthèse "classique" pour le français.

3.5.1.1 Corpus audio

Le corpus d'entraînement est extrait du livre audio : "le tour du monde en 80 jours"[3](précedemment utilisé dans (Bailly et Gouvernayre 2012)), lu par un lecteur non-professionnel. Il s'agit d'un corpus d'une durée totale de 6 heures et 41 minutes comprenant 5 heures et 2 minutes (soit 75% du corpus) de parole et 1 heure et 39 minutes (soit 25% du corpus) de silence. La Figure 3.16 représente le nombre d'occurrences pour chaque classe phonétique dans ce corpus.

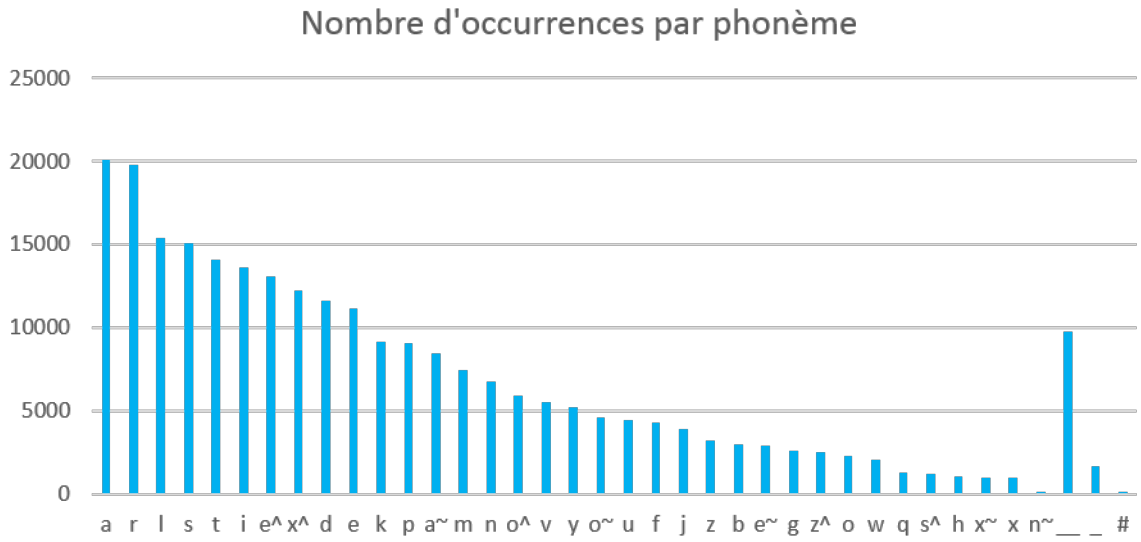


Figure 3.16 – Nombre d'occurrences pour chaque classe phonétique dans le corpus utilisé pour l'entraînement de la voix de synthèse de référence. Les symboles "#", "_" et "__" représentent respectivement les hiatus, les pauses courtes (durée $<$ 200ms) et les pauses longues (durée $>$ 200ms) (voir (Bailly et Gouvernayre 2012) pour l'étude et la modélisation des pauses).

3.5.1.2 Analyse linguistique et segmentation

La phonétisation et l'analyse morpho-syntaxique (contextuelle) ont été réalisées à l'aide du module de TAL de COMPOST (Bailly et Alissali 1992), déjà décrit en Section 2.5.1. Une vérification manuelle a ensuite été effectuée sur l'ensemble des phrases du corpus (notamment pour mettre en cohérence la phonétisation théorique avec ce qui a réellement été prononcé par le locuteur). La segmentation temporelle au niveau phonétique du corpus est obtenue par alignement forcé, à l'aide d'un jeu de modèles HMM de type triphone, estimé sur une autre base de données.

3. Le fichier audio original est disponible à l'adresse suivante : `http://www.litteratureaudio.com/livre-audio-gratuit-mp3/jules-verne-le-tour-du-monde-en-80-jours.html`

3.5.1.3 Paramétrage acoustique

L'extraction des paramètres acoustiques a été réalisée grâce à l'approche harmonique plus bruit (HNM, voir Section 3.3.4). La taille de la fenêtre d'analyse est fixée à 25ms, la fréquence d'analyse à 200 Hz (i.e une observation acoustique est extraite toute les 5 ms), et l'ordre des modèles LPC pour les parties bruit et harmonique respectivement à 16 et 12.

Le paramétrage de la fréquence fondamentale s'effectue à l'aide de l'approche par ondelettes continues présentée à la Section 3.3.6.4 en considérant les quatre bandes de fréquences suivantes : [0.03Hz – 0.7Hz], [1Hz – 3.9Hz], [5.4Hz – 18Hz], [21Hz – 80Hz].

Une observation acoustique est donc un vecteur de 102 paramètres :

— 16 coefficients LSF décrivant l'enveloppe de la composante "bruit" , ainsi que leurs dérivées premières et secondes.
— 12 coefficients LSF décrivant l'enveloppe de la composante "harmonique" , ainsi que leurs dérivées premières et secondes.
— 1 paramètre décrivant le gain de la partie bruit, ainsi que sa dérivée première et seconde.
— 1 paramètre décrivant le gain de la partie harmonique, ainsi que sa dérivée première et seconde.
— 4 paramètres décrivant les transformée en ondelettes continues du logarithme de la fréquence fondamentale (4 bandes) ainsi que leurs dérivées première et seconde.

3.5.1.4 Topologie des modèles HMM

Chaque phonème en contexte est modélisé par un HSMM , de type gauche-droite, à 5 états émetteurs, sans saut d'état. Il s'agit d'un modèle à 6 flux décrivant respectivement :

— La partie harmonique du modèle HNM.
— La partie bruit du modèle HNM.
— Les variations de la fréquence fondamentale dans la bande [0.03Hz – 0.7Hz] (noté $f_0^{b_1}(t)$ par la suite)
— Les variations de la fréquence fondamentale dans la bande [1Hz – 3.9Hz] (appelé $f_0^{b_2}(t)$)
— Les variations de la fréquence fondamentale dans la bande [5.4Hz – 18Hz] (noté $f_0^{b_3}(t)$)
— Les variations de la fréquence fondamentale dans la bande [21Hz – 80Hz] (noté $f_0^{b_4}(t)$)

Les différentes sous-bandes de fréquence de la fréquence fondamentale, $f_0^{b_1}$ à $f_0^{b_4}$, sont modélisées grâce à des flux séparés afin de permettre la construction de 4 arbres de décision pour le regroupement d'états différents. L'approche *Multi-Space Probability Distribution* décrite à la Section 3.3.6.4 est utilisée pour modéliser les paramètres de la partie harmonique, qui ne sont définis que sur les régions voisées du signal.

3.5.1.5 Calcul des valeurs par défaut

Afin d'utiliser la méthode proposée par (Baumann 2014), nous calculons, grâce au modèle non-incrémental, des valeurs de descripteurs contextuels "par défaut" à substituer à une valeur lorsqu'elle est inconnue lors de la synthèse. Comme décrit en Section 3.4.1.3, ces valeurs

TABLE 3.2 – Valeurs par défaut utilisées pour la synthèse incrémentale de la parole selon la stratégie proposée par (BAUMANN 2014)

Phonème suivant	[a]
Phonème suivant-suivant	[a]
Position (comptage rétrogressif) de la syllabe dans la phrase	20
Nombre de phonèmes dans la syllabe suivante	2
Position (comptage rétrogressif) du mot courant dans la phrase	14
Type de phrase	Déclarative
Classe lexicale du mot suivant	Nom
Nombre de syllabes dans le mot suivant	1

par défaut sont calculées à partir du corpus d'entraînement en prenant tous les nœuds de l'arbre dont la question porte sur le contexte droit et en considérant les valeurs de tous les labels interrogés pour ce nœud. Le tableau 3.2 indique la valeur par défaut pour chacun des descripteurs du contexte droit pouvant être inconnu lors de la synthèse incrémentale de la parole.

3.5.2 Évaluations objectives de la stratégie proposée pour la synthèse par HMM incrémentale

Au cours cette section, nous allons successivement décrire les protocoles expérimentaux utilisés pour évaluer objectivement les performances du système TTS non-incrémental (décrit à la section précédente), ainsi que des systèmes TTS incrémentaux basés respectivement sur l'approche de Baumann et al. (Section 3.4.1.3, considéré dans ce travail comme l'état de l'art), sur l'approche d'Astrinaki (Section 3.4.1.2) et sur l'approche que nous proposons (stratégie "*joker*", Section 3.4.2).

3.5.2.1 Corpus d'apprentissage et de test

Afin de permettre l'apprentissage des modèles et l'évaluation des différentes stratégies de synthèse incrémentale, le corpus décrit en Section 3.5.1.1 est divisé en un corpus d'apprentissage (90% des phrases du corpus complet) et un corpus de test (10% du corpus). Le corpus d'apprentissage contient donc 3982 phrases pour une durée totale de 6 heures et 1 minute et le corpus de test 443 phrases (prélevé de manière aléatoire sur le corpus total) pour une durée totale de 40 minutes.

3.5.2.2 Mesure de l'apport de chaque descripteur contextuel

Comme expliqué en Section 3.3.6.5, l'étape de partage d'états permet, pour chacun des 5 états d'un HMM et pour chaque flux, de lier des états acoustiquement proches mais partageant des dépendances contextuelles grâce à des arbres de décision binaires. Une analyse des questions posées lors du parcours de ces arbres de décision peut permettre d'analyser l'influence des dépendances contextuelles sur les paramètres acoustiques qui décrivent le signal de parole.

Nous nous proposons donc d'analyser les dépendances contextuelles de chaque flux par l'intermédiaire des arbres de décision du modèle non-incrémental et ceux du modèle incrémental (basé sur la stratégie proposée "Joker"). Pour ce faire, nous analysons le parcours des arbres de décision pour un ensemble de labels (phonème et descripteurs contextuels) issus du corpus de test. Nous proposons plusieurs mesures pour caractériser les dépendances contextuelles des différents flux du modèle non-incrémental et du modèle incrémental.

- Pour les arbres de décision du modèle non-incrémental :
  - Nous proposons de calculer le nombre de questions posées pour chaque élément du contexte droit : Passé – Phone courant – phone suivant – deux phones suivants – syllabe courante – mot courant – syllabe suivante – mot suivant – contexte complet (jusqu'à fin de la phrase).
  - Nous proposons également de compter le nombre de questions nécessitant la connaissance du mot suivant. Comme indiqué dans le Tableau 3.1, certains descripteurs contextuels peuvent dépendre du mot suivant en fonction de leur place dans le mot courant (e.g : si le phonème courant est le dernier du mot courant, le phonème suivant dépend du mot suivant). Nous cherchons à compter le nombre de fois où le descripteur contextuel interrogé dépend du mot suivant.
- Pour les arbres de décision du modèle incrémental (i.e. stratégie "joker") :
  - Nous proposons de compter le nombre de questions interrogeant un descripteur contextuel dépendant du mot suivant, en identifiant les questions portant sur un descripteur inconnu (descripteur "joker") et celles portant sur un descripteur connu.

3.5.2.3 Distorsion Mel-cepstrale

Nous nous proposons également de mesurer la différence de qualité au niveau segmental en comparant la stratégie de référence (*baseline*) de Baumann et al., la stratégie que nous proposons dans ce travail de thèse et la stratégie consistant à supprimer tous les descripteurs contextuels pouvant être inconnus au moment de la synthèse (proposée par (Astrinaki 2014), notée ici "Sans Contexte Droit"). Nous adoptons ici la même hypothèse de travail que (Baumann et Schlangen 2012a) selon laquelle le paradigme classique (par phrase) fournira toujours une meilleur voix de synthèse que le paradigme incrémental. Ainsi, les différentes mesures que nous réalisons se font en confrontant les signaux de synthèse incrémentale aux signaux de synthèse non-incrémentale (et non pas aux signaux originaux).

Afin de comparer les qualités segmentales des signaux de synthèse, nous proposons d'utiliser une mesure de distorsion spectrale rendant compte des propriétés de la perception de

l'oreille humaine : la distorsion mel-cepstrale (voir KUBICHEK 1993). Cette mesure est largement employée dans la littérature pour l'évaluation des systèmes TTS et des systèmes de conversion automatique de la voix.

En s'appuyant sur l'Équation 3.1, nous calculons un jeu de 25 coefficients mel-cepstraux. On note $\mathbf{y}^S(t)$ le vecteur de 25 coefficients mel-cepstraux à la trame t pour le signal synthétisé avec la stratégie de synthèse incrémentale S (avec $S \in$ {"Joker" , "Par Défaut" , "Sans Contexte Droit" }). De même, on note $\mathbf{y}_t^{NI}$ le vecteur de coefficients mel-cepstraux à la trame t pour le signal de synthèse non-incrémental. En notant T le nombre total de trames du signal, la distorsion mel-cepstrale entre les deux signaux se calcule de la façon suivante :

$$MCD(\mathbf{y}^S, \mathbf{y}^{NI}) = \frac{1}{T}\frac{10}{ln(10)}\sum_{t=1}^{t=T}\sqrt{2\sum_{d=0}^{D}(\mathbf{y}^S(t) - \mathbf{y}^{NI}(t))^2} \tag{3.17}$$

Afin de pouvoir comparer les différentes stratégies entre elles, on impose que la durée des phonèmes synthétisées soit égale à la durée originale des phonèmes (tels que prononcés par le locuteur).

3.5.2.4 Fréquence fondamentale

La fréquence fondamentale étant modélisée par plusieurs bandes de fréquence issue de la transformation en ondelettes continues, nous proposons de mesurer la qualité de l'estimation d'une part pour chaque bande de fréquence, et d'autre part sur la courbe de f_0 reconstruite.

Pour estimer l'erreur commise sur chacune des sous bandes de $f_0(t)$, nous proposons d'utiliser deux mesures : la corrélation entre les signaux issus générés par les modèles incrémentaux et le modèle non-incrémental (entre 0 et 1) ainsi que l'erreur quadratique moyenne.

Pour mesurer l'erreur en terme de fréquence fondamentale reconstruite, nous utilisons la mesure proposée par (PERETZ et HYDE 2003) (également utilisée par ASTRINAKI et al. 2012). Cette mesure s'exprime en cents ; un écart de 100 cents correspond à un demi-ton et un écart de 25 cents est considéré comme inaudible. Elle est notée E_{f_0} et se calcule à l'aide de la formule suivante :

$$E_{f_0} = \frac{1}{T}\sum_{t=1}^{T} 1200 \log_2 \left| \frac{f_0^S(t)}{f_0^{NI}(t)} \right| \tag{3.18}$$

Ici aussi, pour la mesure d'erreur en terme de fréquence fondamentale, la durée des phonèmes de synthèse est fixée à la durée originale des phonèmes.

3.5.2.5 Durées

L'erreur commise sur la durée des phonèmes induite par l'utilisation d'une approche incrémentale est définie telle que :

$$E_{dur} = \frac{1}{P}\sum_{p=1}^{P} \log d_p^{NI}/d_p^S \tag{3.19}$$

avec d_p^{NI} la durée du p^{eme} phonème (sur un total de P phonèmes) en synthèse classique et d_p^S la durée de ce même phonème en synthèse incrémentale.

3.5.2.6 Signification statistique

La signification statistique de chaque mesure (MCD, E_{f_0}, E_{dur}) est estimée à l'aide de test de Student apparié. Le test de signification (ici pour la distorsion mel-cepstrale) permet de donc de rendre compte de la différence suivante : $MCD_{S_i,NI}$ vs $MCD_{S_j,NI}$.

3.5.3 Résultats

3.5.3.1 Propriétés des voix de synthèse

L'exploitation du contexte gauche et droit pour la synthèse (non-incrémental) par HMM du français est illustrée à la Figure 3.17. Pour chaque niveau de contexte, nous représentons, en pourcentage, le nombre de nœuds (et donc de questions) de l'arbre de décision binaire de la procédure de partage d'état qui s'y rapporte, pour chacun des différents flux de paramètres acoustiques. On considère les 9 catégories suivantes : contexte gauche – phonème courant – phonème suivant – phonème suivant le phonème suivant – syllabe courante – mot courant – syllabe suivante – mot suivant – contexte complet.

Nous pouvons constater que l'inférence des paramètres spectraux (partie harmonique et bruit du modèle HNM) nécessite principalement de connaître le quinphone. En effet pour chacun de ces deux paramètres, la connaissance du quinphone suffit pour répondre à 95% des questions posées. Ce constat est en adéquation avec les résultats de (Le Maguer 2013). En ce qui concerne la fréquence fondamentale, on peut constater que les bandes hautes fréquences ($f_0^{b_3}$ et $f_0^{b_4}$) dépendent principalement du quinphone tandis que les bandes basses fréquences ($f_0^{b_1}$ et $f_0^{b_2}$) exploitent un empan temporel plus large (mot courant, mot suivant et descripteurs contextuels nécessitant la connaissance de la fin de l'énoncé pour être calculés).

Dans le cadre de la synthèse incrémentale, le niveau de granularité étant celui du mot, nous cherchons également à calculer le pourcentage de descripteurs interrogés appartenant au mot suivant, quelque soit le niveau linguistique interrogé (phonème, syllabe, etc.). La Figure 3.18a indique le pourcentage de questions posées nécessitant la connaissance d'un élément appartenant au mot suivant (ou tout autre mot du contexte droit), pour chacun des 5 états HMM, et pour chaque flux de paramètres acoustiques. La Figure 3.18b indique le pourcentage de feuilles des différents arbres atteint en ayant interrogé au moins un descripteur contextuel appartenant au mot suivant.

Ces figures mettent en évidence l'influence des éléments appartenant au mot suivant lors de phase de sélection d'états. Nous pouvons constater que même pour les paramètres acoustiques

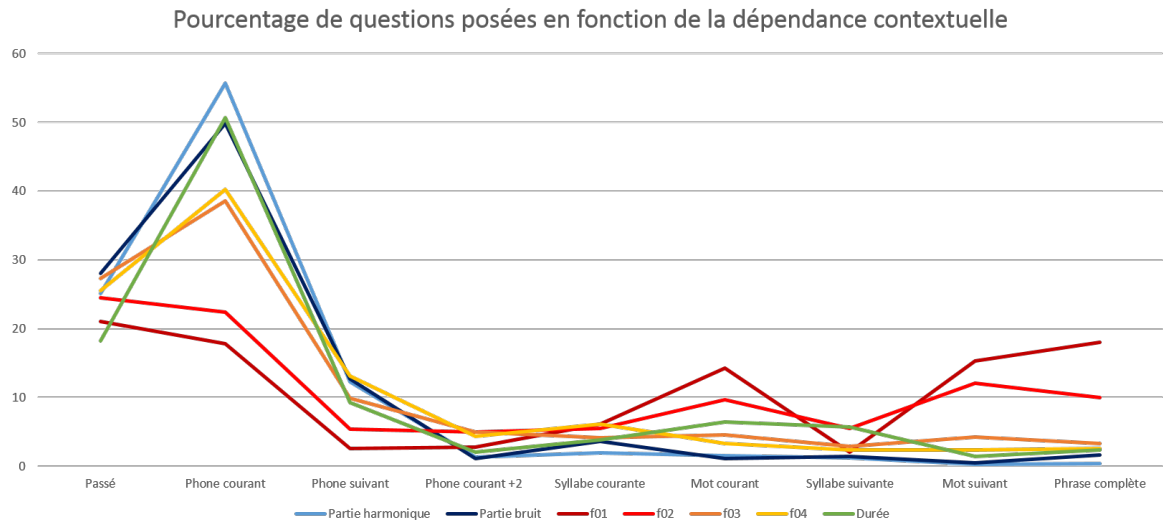
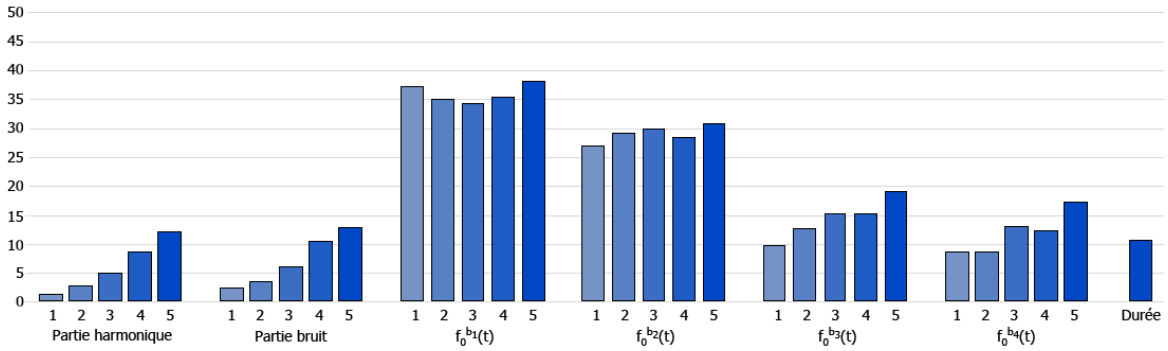


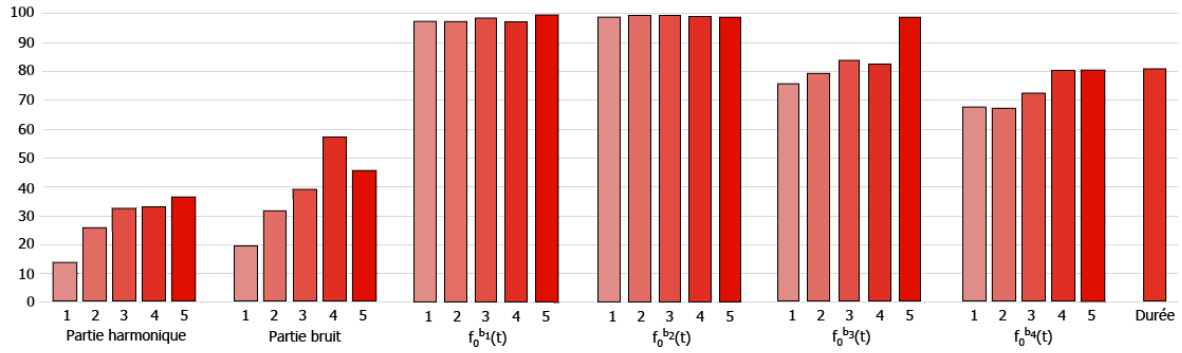
FIGURE 3.17 – Exploitation du contexte gauche et droit pour le partage des états HMM dans le cas de la synthèse du français, pour les différents flux de paramètres acoustiques (en bleu clair et foncé les parties harmonique et bruit de la modélisation HNM, en jaune, orange, rouge et bordeaux, les 4 premiers coefficients de la décomposition en ondelettes continues de la fréquence fondamentale, en vert, la durée des phonèmes). Représentation inspirée de (BAUMANN 2014).

segmentaux (c'est-à-dire les parties harmonique et bruit du modèle HNM et les variations rapides de la fréquence fondamentale, encodées principalement par $f_0^{b_3}$ et $f_0^{b_4}$) au moins 30% des états sont sélectionnés en interrogeant au moins une fois un élément du mot suivant. Pour les paramètres suprasegmentaux, cette quantité monte à 80% pour la durée et près de 100% pour les variations lentes de la fréquence fondamentale.

La stratégie de référence de Baumann et al. (stratégie "Par Défaut") décrite précédemment, est directement impactée par ces résultats. En effet, lors du parcours des arbres de décision par les labels, à chaque fois qu'un élément contextuel appartenant au mot suivant est interrogé (et donc dont la valeur est manquante en synthèse incrémentale), la réponse à la question est imposée. Comme le montre la Figure 3.18a, si les effets de cette stratégie sur la qualité segmentale peuvent être considérés comme négligeables, les répercussions sur la qualité suprasegmentale risquent d'être plus importantes (cela se confirmera par nos évaluations objectives et subjectives présentées en Sections 3.5.3.2 et 3.5.4). On constate en effet qu'entre 30 et 40% des questions posées pour la sélection des états, pour les flux $f_0^{b_1}$ et $f_0^{b_2}$ (c'est-à-dire les variations lentes, à l'échelle du syntagme ou au delà) se rapportent au mot suivant. Par ailleurs, nous pouvons constater que pour les parties harmoniques et bruit, du modèle HNM ainsi que pour $f_0^{b_3}$ et $f_0^{b_4}$, plus on approche de l'état modélisant la fin du phonème (5^{eme} état), plus l'inférence des paramètres acoustiques nécessite la connaissance du mot suivant. À l'inverse, cette évolution n'est pas ou peu visible sur les états des HMM modélisant les variations lentes de la fréquence fondamentale ($f_0^{b_1}$ et $f_0^{b_2}$), tendant à montrer que des informations plus



(a) Pourcentage de questions interrogeant un élément du mot suivant lors de la phase de sélection d'états pour la synthèse non-incrémentale pour chaque état HMM (numérotés de 1 à 5), et pour chaque flux de paramètres acoustiques

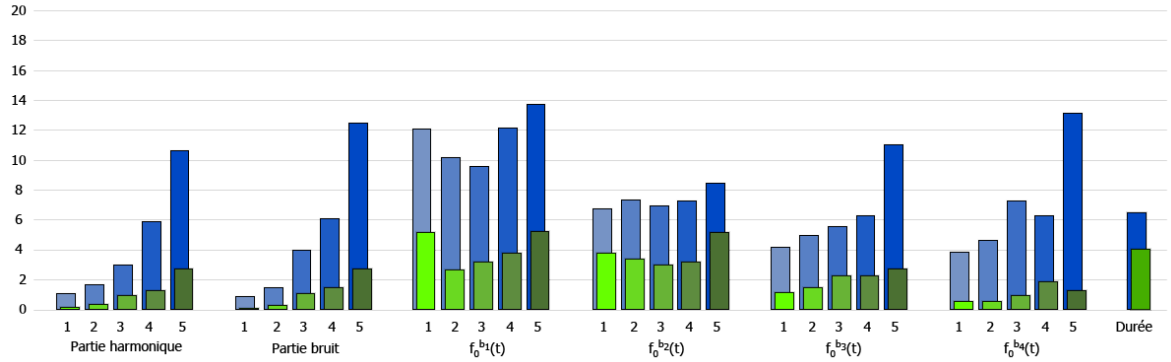


(b) Pourcentage d'états dont le calcul nécessite la connaissance d'au moins un élément appartenant au mot suivant lors de la phase de sélection d'états pour la synthèse non-incrémentale pour chaque état HMM, et pour chaque flux de paramètres acoustiques

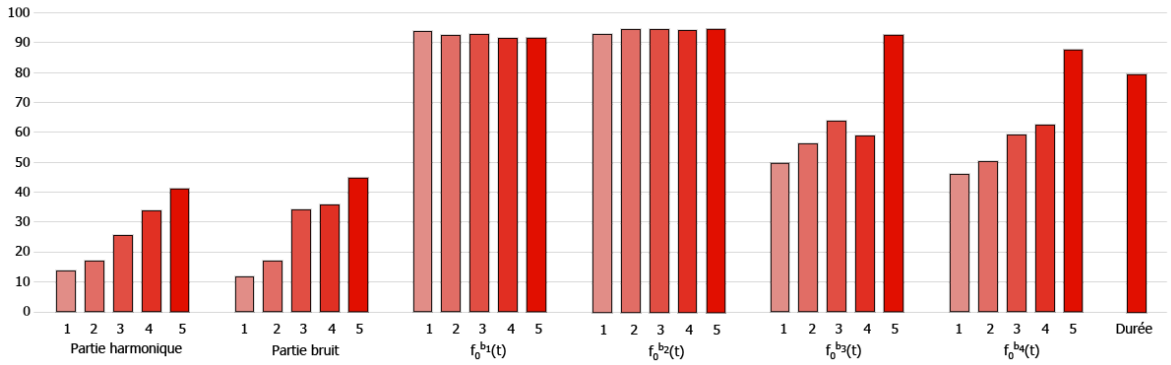
FIGURE 3.18 – Pourcentage de question pour chacun des arbres de décision (chaque flux et chacun des 5 états) portant sur le contexte droit pour le modèle non-incrémental

globales sont requises.

Enfin, les Figures 3.19a et 3.19b indiquent respectivement les pourcentages de questions et d'états dont le calcul a nécessité au moins un élément contextuel dépendant du mot suivant pour la synthèse incrémentale basée sur la stratégie proposée ("Joker").



(a) Pourcentage de questions interrogeant un élément du mot suivant lors de la phase de sélection d'états pour la synthèse incrémentale (stratégie "Joker") du corpus de test en fonction du flux et de l'état (numérotés de 1 à 5) du HMM (en bleu). En vert, est indiqué, par état et par flux, le pourcentage de question portant sur un "Joker" (par exemple : "Le nombre de phonèmes dans la syllabe de droite est il connu ?").



(b) Pourcentage d'états dont le calcul nécessite la connaissance d'au moins un élément appartenant au mot suivant lors de la phase de sélection d'états pour la synthèse incrémentale (stratégie "Joker") du corpus de test en fonction du flux et de l'état (numérotés de 1 à 5) du HMM

FIGURE 3.19 – Pourcentage de question pour chacun des arbres de décision (chaque flux et chacun des 5 états) portant sur le contexte droit pour le modèle incrémental (stratégie "Joker")

Nous pouvons constater grâce à ces résultats que, dans le cas du modèle "Joker", utilisant les descripteurs contextuels indisponibles comme une information pertinente pour la modélisation, beaucoup moins de questions portant sur le mot suivant sont posées (14% maximum, contre près de 38% pour le modèle non-incrémental). Ces descripteurs sont cependant toujours utilisés

car, lorsqu'ils sont porteurs d'informations pertinentes (i.e. lorsque la valeur du descripteur contextuel est connue), celles-ci doivent être exploitées. Parmi les questions portant sur le mot suivant, notons également que, pour les paramètres acoustiques $f_0^{b_1}$, $f_0^{b_2}$ et "durée", une grande proportion de ces questions consiste à se demander si le paramètre en question est connu lors du traitement incrémental. Ces constats nous permettent de conclure (a) que la solution proposée semble plus adaptée à la synthèse incrémentale. En effet, les questions recrutées lors de la phase de *state-tying* du modèle incrémental s'appuient beaucoup moins sur le contexte droit que lors du state-tying du modèle non-incrémental. (b) nous constatons qu'une importante proportion des questions portant sur le contexte droit portent sur le caractère déterminé ou non du descripteur contextuel (Figure 3.19a). L'introduction d'un descripteur contextuel "joker" semble donc jouer un rôle important dans le partitionnement de l'espace acoustique dans le cadre de la synthèse incrémentale.

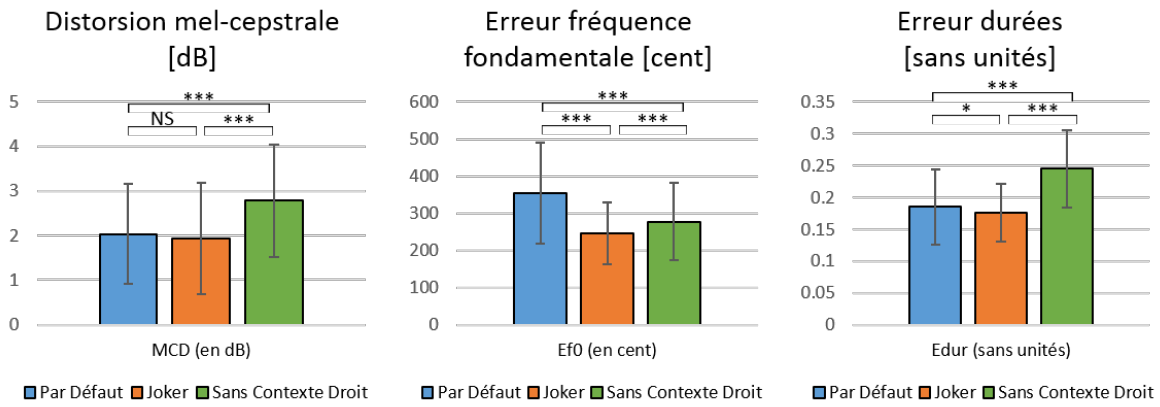
3.5.3.2 Mesures acoustiques

Les résultats de l'évaluation objective "synthèse HMM classique *versus* synthèse HMM incrémentale", au niveau segmental (MCD) et suprasegmental (f_0, durée), et pour les trois stratégies incrémentales étudiées (la stratégie "Par Défaut" de Baumann et al., "Sans Contexte Droit", d'Astrinaki et al., et la stratégie proposée "Joker"), sont présentés aux Figures 3.20a et 3.20b.

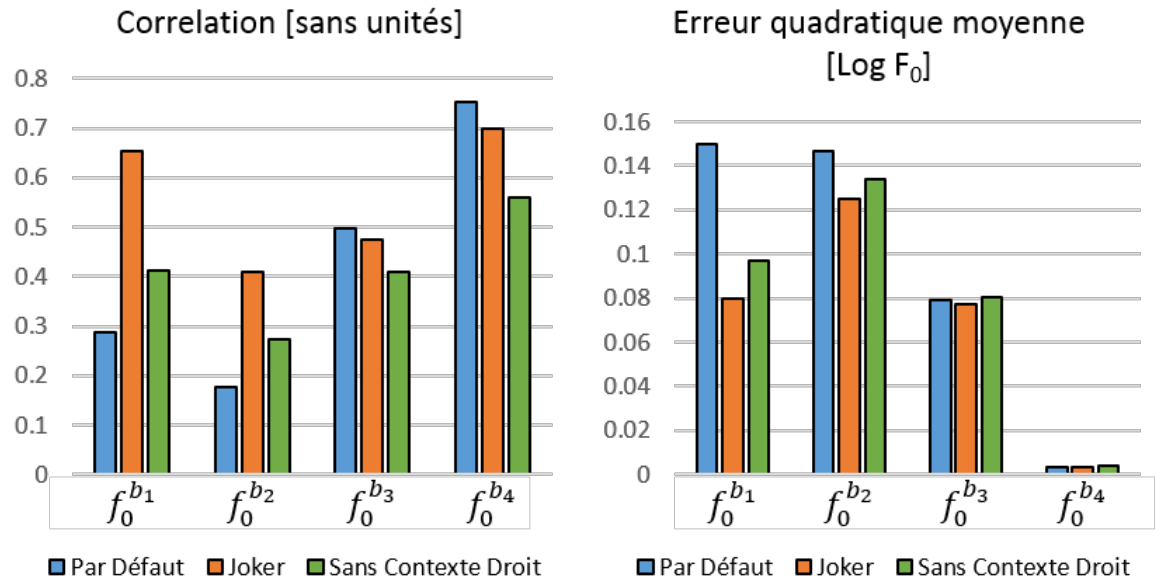
Nous constatons tout d'abord que les stratégies "Par Défaut" et "Joker" sont équivalentes en terme de qualité segmentale (différence non significative en terme de distorsion mel-cepstrale). La stratégie consistant à supprimer le contexte droit, en revanche, subit une dégradation de presque un décibel supplémentaire par rapport à la synthèse non-incrémentale. Cette absence de différence significative entre les stratégies "Par Défaut" et "Joker" pouvait être attendu étant donné la faible exploitation du contexte droit pour l'estimation des paramètres spectraux en synthèse par HMM, tel que montré, à la fois par Le Maguer et al, et par nos analyses présentées à la Section 3.5.3.1. À l'inverse, lorsque tous les descripteurs portant sur le contexte droit sont supprimés (et donc les deux phonèmes suivant celui à synthétiser), une dégradation de la qualité segmentale du signal peut être attendue.

Au niveau suprasegmental, nous constatons une meilleure qualité d'estimation de la f_0 et des durées pour la stratégie proposée "Joker". C'est notamment le cas pour les mesures E_{dur} (0.17 pour la stratégie "Joker" contre 0.18 et 0.24 pour les stratégies "Par Défaut" et "Sans Contexte Droit"), E_{f_0} globale (247 cents contre 354 et 278), et les variations lentes de f_0 ($f_0^{b_1}$, $f_0^{b_2}$), relatives aux macro-variations prosodiques (c'est-à-dire celles que l'on peut observer à l'échelle du syntagme et/ou de la phrase). De façon assez surprenante, l'erreur en terme de fréquence fondamentale (que ce soit pour la reconstruction ou pour la transformée en ondelettes continues) est plus faible lorsqu'elle est inférée sans connaissances sur le contexte droit qu'avec une substitution des valeurs inconnues par une valeur par défaut.

Ces résultats laissent donc suggérer une meilleure qualité prosodique pour la stratégie proposée "Joker". Nous cherchons à présent à valider ces résultats à l'aide de tests perceptifs.



(a) Évaluation objective des stratégies incrémentales "Par Défaut" (en bleu), "Joker" (en orange) et "Sans Contexte Droit" (vert) par rapport à la synthèse non-incrémentale (considérée ici comme la référence). Les erreur sont mesurées en terme de distorsion mel-cepstrale (MCD), de fréquence fondamentale (E_{f_0}) et de durée (E_{dur}). Les barres d'erreurs indiquent l'écart-type pour chaque mesure.



(b) Erreur entre les différentes bandes de fréquence ($f_0^{b_1}$ à $f_0^{b_4}$) de la fréquence fondamentale en terme de corrélation et d'erreur quadratique moyenne par rapport à la synthèse non-incrémentale

FIGURE 3.20 – Mesures acoustiques des erreurs entre les signaux de synthèse incrémentale ("Par Défaut", "Joker" et "Sans Contexte Droit") et les signaux de synthèse non-incrémentale. Les mesures de corrélation et d'erreur quadratique moyenne sont réalisées sur chacune des sous-bandes de fréquences de f_0 synthétisée par les modèles.

3.5.4 Évaluation perceptive

3.5.4.1 Protocole expérimental

Le test subjectif proposé vise à faire évaluer par un ensemble d'auditeurs une même phrase synthétisée selon trois stratégies : les stratégies incrémentales "Par Défaut" et "Joker", et l'approche non-incrémentale classique. Nous choisissons volontairement de ne pas évaluer perceptivement la stratégie "Sans Contexte Droit" pour plusieurs raisons : d'une part afin de limiter la durée totale du test, d'autre part car une première écoute des stimuli générés par cette stratégie indique une qualité largement inférieure. Afin de ne pas avoir une trop grande démarcation entre les stimuli générés grâce à la stratégie "Sans Contexte Droit" et les autres, nous choisissons de ne pas l'intégrer à l'évaluation perceptive. Pour l'évaluation, 12 phrases ont été aléatoirement extraites du corpus de test. Les phrases utilisées sont fournies en annexe (B)

L'ensemble de ces 12 phrases synthétisées selon trois stratégies représentent un corpus de 36 stimuli. Le test à été soumis à 18 sujets francophones naïfs (c'est-à-dire sans aucune expertise particulière en synthèse de la parole ou en linguistique), auxquels il a été demandé de classer les stimuli "du moins naturel au plus naturel". Le test a été réalisé dans un environnement silencieux à l'aide d'un casque audio. La Figure 3.21 est une capture d'écran de l'interface développée pour la présentation des stimuli de ce test.

De façon similaire à (Pfitzinger 1998) et (Bailly et Gorisch 2006), il est demandé au sujet de classer sur un même axe les trois versions d'une même phrase. Cette méthode lui permet de classer les stimuli et d'affiner ce classement en attribuant des scores proches aux stimuli qui diffèrent peu et des scores plus distants s'il souhaite marquer une plus grande différence entre deux stimuli. Le sujet a pour tâche d'écouter les stimuli numérotés de 1 à 3 en double-cliquant sur la vignette correspondante (la vignette en cours d'écoute devient rouge, comme montré sur la partie inférieure de la figure), puis de placer la vignette sur la grille (deux dimensions) de notation. L'abscisse de cette grille est un axe continu, sur lequel nous avons cependant fait figurer cinq labels qualitatifs ("Très Mauvais", "Mauvais", "Moyen", "Bon", et "Excellent") pour donner quelques repères au sujet (ceci s'est avéré utile principalement au début du test).

Il était précisé dans les instructions (et rappelé à chaque présentation d'un nouveau stimulus) que la position verticale n'a pas d'importance. L'axe vertical permet cependant de superposer plusieurs stimuli jugés proches. Avant de pouvoir déplacer une vignette, le sujet doit d'abord double-cliquer une première fois dessus pour l'écouter. Il est précisé au sujet qu'il peut ré-écouter chaque stimulus autant de fois qu'il le souhaite. Lorsqu'il a fini de classer les stimuli, et qu'il clique sur "Valider", les 4 stimuli sont rejoués du moins bien classé au mieux classé. Pour chaque sujet, on applique une permutation aléatoire sur l'ordre des phrases. De même, pour chaque phrase, les stimuli sont aléatoirement placés derrière les vignettes.

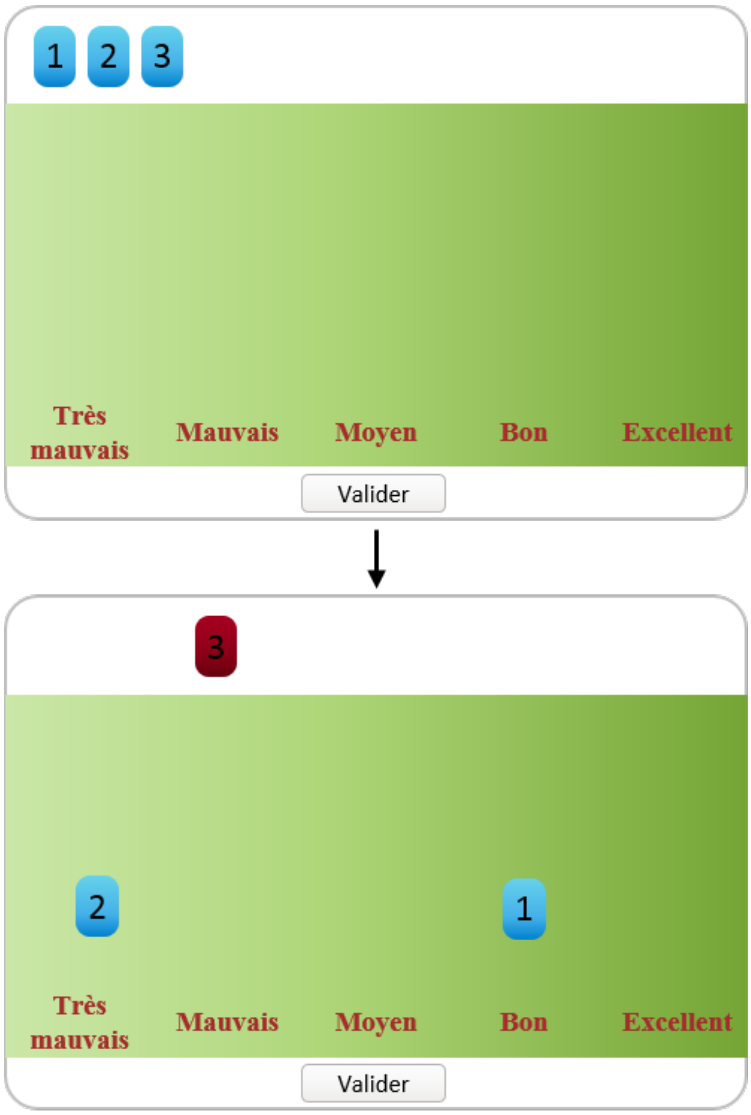


FIGURE 3.21 – Interface développée pour la présentation des stimuli du test d'évaluation perceptive. Sur la partie supérieure de l'image, la grille avant que l'utilisateur n'ait commencé le classement, sur la partie inférieure, la grille durant l'évaluation. Les vignettes numérotées correspondent aux stimuli présentés avec, en rouge, un stimulus en cours de lecture.

3.5.4.2 Méthode d'analyse des résultats

L'analyse des résultats s'effectue à l'aide d'une analyse statistique de type "régression bêta" (Ferrari et Cribari-Neto 2004). Cette analyse se base sur le modèle linéaire suivant :

$$\log \frac{y}{1-y} = \mu + \alpha_i + \tau_s + \tau_p \tag{3.20}$$

avec :

— μ : l'effet global moyen
— α_i : un effet fixe décrivant l'influence des stratégies sur la variable réponse y.
— τ_s : un effet aléatoire décrivant l'influence des sujets sur la variable réponse y.
— τ_p : un effet aléatoire décrivant l'influence des phrases sur la variable réponse y.

Les méthodes de comparaisons multiples en cas de significativité des résultats sont décrites dans (Hothorn, Bretz et Westfall 2008). L'analyse statistique a été réalisée sous le logiciel R, à l'aide du paquet `glmmADMB`.

Les résultats indiquant qu'il y a un effet significatif du facteur "stratégie", nous pouvons donc réaliser des tests post-hoc (et comparer ainsi les stratégies 2 à 2). Les résultats sont présentés à la Figure 3.22.

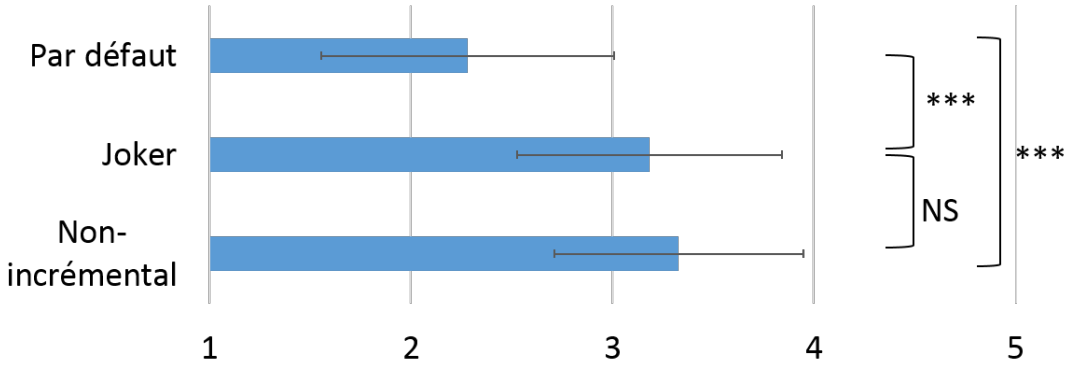


Figure 3.22 – Résultats de l'évaluation perceptive : position moyenne des stimuli regroupés par stratégie. Sur la figure, NS signifie qu'il n'y a pas de différence significative entre les distribution et *** signifie une différence très significative ($p < 0.005$).

3.5.4.3 Interprétation

Comme attendu, la méthode de synthèse qui obtient les meilleurs résultats est la synthèse non-incrémentale (score moyen de 3.32), que l'on a considérée comme cible lors des mesures objectives. De façon plus surprenante, nous n'observons aucune différence significative entre la synthèse incrémentale basée sur la stratégie proposée "Joker", et la synthèse non-incrémentale.

Ce résultat semble démontrer la pertinence de la stratégie proposée qui permet presque d'obtenir, en synthèse incrémentale, la même qualité qu'en synthèse classique. Il faut toutefois nuancer cette interprétation par la qualité intrinsèque du synthétiseur par HMM du français

que nous avons mis en place lors de cette thèse. En effet, lors de l'évaluation perceptive, nos synthèses "classiques" ont été jugées en moyenne seulement à 3.3, soit entre "Moyen" et "Bon". Cette qualité relative peut être en partie attribuée à la technique de synthèse utilisée (l'approche par HMM) dont les limitations commencent aujourd'hui à être décrites dans la littérature (cette technique laisse notamment aujourd'hui sa place aux approches par réseaux de neurones récurrents, comme décrit dans (Zen, Senior et Schuster 2013,Zen et Sak 2015)). L'approche proposée consistant à intégrer une incertitude sur le contexte droit pourrait cependant être appliquée à cette modélisation.

Cependant, l'évaluation perceptive valide les conclusions de l'évaluation objective en montrant une supériorité de l'approche proposée (stratégie 'Joker') par rapport à l'approche état-de-l'art de Baumann et al. (stratégie "Par Défaut").

3.6 Conclusions et perspectives

Dans ce chapitre, nous avons tout d'abord décrit le principe et les différentes étapes nécessaires à la mise en place d'un synthétiseur vocal paramétrique basé sur une modélisation par HMM. Après avoir identifié les verrous à lever pour adapter cette approche au paradigme de la synthèse incrémentale et présenté l'état de l'art sur ce sujet, nous avons décrit une adaptation de la procédure d'entraînement de la voix de synthèse par HMM. La méthode proposée consiste 1) à entraîner des modèles HMM pour des contextes potentiellement incomplets, et 2) à lier les modèles partageant la même incertitude sur des descripteurs contextuels liés au contexte droit. L'approche proposée est comparée à l'approche de référence proposée par (Baumann 2014). Cette dernière est basée sur l'utilisation, au moment de la synthèse, d'une valeur par défaut pour chaque descripteur contextuel manquant lors du traitement incrémental. Des évaluations objectives et perceptives ont montré l'intérêt de l'approche proposée (pour le français).

Dans le futur, des travaux pourront porter sur la validation de l'approche proposée pour d'autres langues telles que l'anglais ou l'allemand, qui sont les langues considérées dans les travaux de Baumann et al. . Par ailleurs ces langues ayant une accentuation plus marquées que le français, l'étude de la prosodie inférée dans un contexte incrémental par des HMM pourrait être intéressante.

Enfin, dans le cadre de la synthèse incrémentale de parole à partir de corpus audio, une autre approche pourrait consister à construire un corpus de parole dont le locuteur découvre le texte au moment de l'enregistrement (et ce de façon progressive). Tandis que l'approche proposée permet d'avoir une cohérence entre données (textuelles) d'apprentissage et données de test, un tel corpus permettrait également d'avoir une cohérence entre données textuelles et observations acoustiques lors de l'apprentissage.

Chapitre 4

Prototype de synthétiseur incrémental de parole à partir du texte.

Sommaire

4.1 Introduction

Nous avons présenté au cours des chapitres précédents plusieurs contributions pour l'adaptation de la chaîne de traitement classique d'un système TTS au paradigme de la synthèse incrémentale. Ces dernières concernent respectivement le module de traitement automatique du langage naturel (TAL), et le module de synthèse sonore (basé dans notre cas sur une approche paramétrique par HMM). Les différentes méthodes proposées ont été couplées afin de réaliser un prototype complet d'un synthétiseur *Text-to-Speech* incrémental en français.

La présentation de ce prototype fait l'objet de ce chapitre qui est divisé en trois sections au cours desquelles nous décrirons :

— La technique proposée pour le couplage de l'analyseur morpho-syntaxique "à latence adaptative" décrit au Chapitre 2 et la technique de construction d'une voix de synthèse par HMM adaptée à la synthèse incrémentale décrite au chapitre 3.
— Les 2 versions de ce prototype basées respectivement sur un logiciel autonome (*standalone*) sur PC, et sur une architecture client-serveur à destination des tablettes et smartphones.
— Une première évaluation perceptive du prototype développé dans cette thèse.

4.2 Méthodologie

4.2.1 Fonctionnement général du système TTS incrémental proposé

La figure 4.1 présente le fonctionnement général du système TTS incrémental proposé, issu du couplage de l'analyseur morpho-syntaxique "à latence adaptative" (chapitre 2) et de la synthèse par HMM incrémentale (chapitre 3).

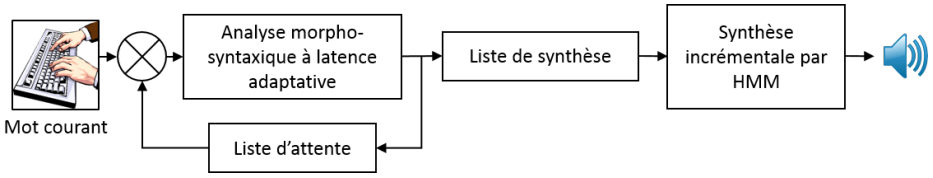


FIGURE 4.1 – Fonctionnement général du système TTS incrémental proposé. À l'issue de l'analyse morpho-syntaxique, le mot saisi est soit placé dans une liste d'attente, pour augmenter la taille de son contexte, soit placé dans une liste de synthèse (potentiellement avec d'autres mots qui étaient dans la liste d'attente) afin d'être synthétisé.

Le module de TAL est invoqué lors de la saisie d'un nouveau mot par l'utilisateur (détecté par la saisie d'un espace ou d'une ponctuation), D'après notre technique d'analyse morpho-syntaxique "à latence adaptative", la décision de synthétiser ce mot est conditionnée par la stabilité de sa classe lexicale. Cette dernière est estimée en ligne, à l'aide d'un ensemble de 3 arbres de décisions binaires, entraînés sur un large corpus d'apprentissage, et considérant un contexte droit de 0, 1 ou 2 mots.

Aussi, dans le système proposé, la parole de synthèse est susceptible d'être délivrée par groupes de 1 à 4 mots. Pour chaque mot d'un tel groupe, la taille du contexte droit disponible est donc variable (exemple, pour décrire le premier mot d'un groupe de 3 mots on peut exploiter 2 mots de contexte droit). Contrairement au chapitre précédent dans lequel nous nous placions dans le cas de l'absence de contexte droit, nous proposons ici une méthode de synthèse permettant d'exploiter au mieux ces différentes tailles au sein d'un même groupe de mots.

Nous proposons d'utiliser conjointement plusieurs modèles de synthèse (i.e. un ensemble de modèles HMM contextuels), tous entraînés sur le même corpus à l'aide de la stratégie proposée "Joker", mais en considérant pour chacun un contexte droit différent (de 0, 1, 2 ou 3 mots). La procédure d'inférence des paramètres acoustiques à l'aide de ces quatre modèles est illustrée à la figure 4.2.

Le HMM modélisant le groupe de mots à synthétiser est construit par concaténation des différents phonèmes (en contexte) de chacun des mots. Les trajectoires de paramètres acoustiques sont ensuite inférées à l'aide de l'algorithme MLPG, mais en considérant une suite d'états HMM provenant (potentiellement) de différents modèles de synthèse (notés ici L_0, L_1, L_2 et L_3, pour 0 à 3 mots de contexte droit).

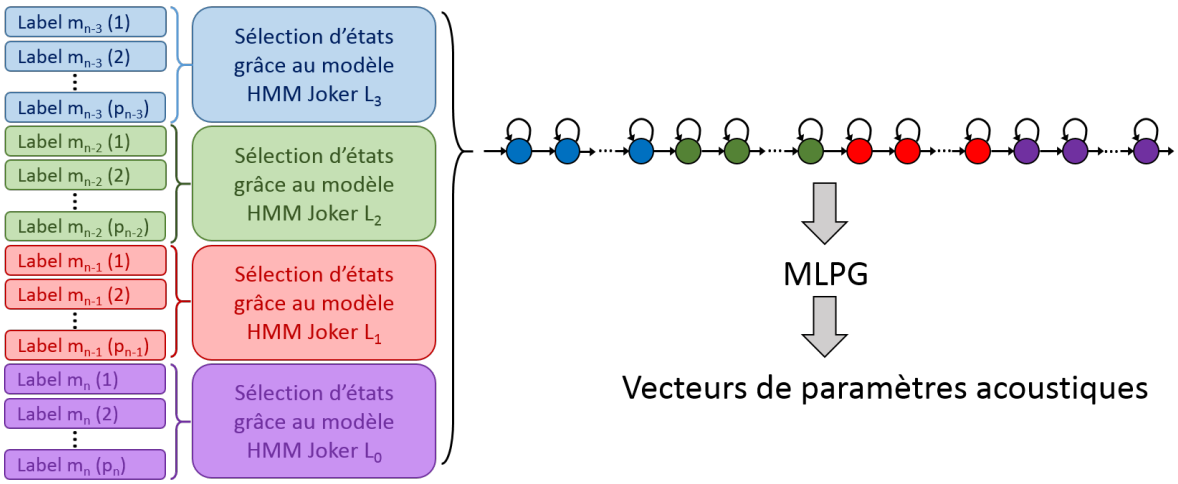


FIGURE 4.2 – Construction du HMM du groupe de mots à synthétiser par sélection d'états issus de modèles de synthèse entraînés avec un contexte droit variable. Chaque mot m_i est composé de p_i phonèmes en contexte (avec un contexte droit de taille variable, de 0 à 3 mots, représenté par différentes couleurs), chacun représenté par un label.

L'algorithme complet de synthèse TTS incrémentale proposé, résultant du couplage de l'analyse morpho-syntaxique à latence adaptative et de la procédure d'inférence de paramètres acoustiques exploitant différents modèles de synthèse (entraînés avec un contexte droit de taille variable) est décrit (en pseudo-code) à l'Algorithme 3).

Data:
- Les 3 derniers mots entrés $[m_{i-2}, m_{i-1}, m_i]$ et les classes lexicales inférées incrémentalement $[c_{t-2}, c_{t-1}, c_t]$
- Une liste d'attente *waiting list* contenant les mots dont la classe lexicale n'a pas été estimée comme étant stable. En notant m_i le dernier mot tapé, *waiting list* peut potentiellement contenir $[m_{i-3}, m_{i-2}, m_{i-1}]$
- Une liste de synthèse *synthetize list* contenant l'ensemble des mots à synthétiser (ainsi que le modèle à utiliser pour la synthèse).
- 4 modèles de synthèse HMM L_0, L_1, L_2 et L_3 entraînés à l'aide de la stratégie 'Joker' en considérant différentes tailles de contexte droit (respectivement de 0 à 3 mots).

if m_{t-3} *is in waiting list* **then**
  put (m_{t-3}, L_3) in *synthetize list*
  `/* Un mot est systématiquement synthétisé après 3 mots d'attente, à l'aide du modèle` L_3 `*/`

if m_{t-2} *is in waiting list* **then**
  if *IsStable*(c_{t-2}) *(Contexte droit = 2 mots)* **then**
    put (m_{t-2}, L_2) in *synthetize list*
  else
    Put m_{t-2}, m_{t-1}, m_t in *waiting list* ;
    return ;

if m_{t-1} *is in waiting list* **then**
  if *IsStable*(c_{t-1}) *(Contexte droit = 1 mot)* **then**
    put (m_{t-1}, L_1) in *synthetize list*
  else
    Put m_{t-1}, m_t in *waiting list* ;
    return ;

if m_t *is in waiting list* **then**
  if *IsStable*(c_t) *(Contexte droit = 0 mot)* **then**
    put (m_t, L_0) in *synthetize list*
  else
    Put m_t in *waiting list* ;
    return ;

synthetize(*synthetize list*) **return** ;

Algorithm 3: Algorithme complet pour la synthèse TTS incrémentale par HMM, basé sur l'algorithme d'analyse morpho-syntaxique à latence adaptative et sur un ensemble de modèles de synthèse par HMM entraînés à l'aide de la stratégie 'Joker' en considérant différentes tailles de contexte droit (entre 0 et 3 mots).

4.3 Description des prototypes

4.3.1 Logiciel autonome iTTS

Le premier prototype est basé sur une architecture de type logiciel autonome (appelé *standalone*), à destination d'un environnement de type PC. Une capture d'écran de son interface utilisateur est fournie en Figure 4.3.

Le moteur de cette application est basé sur une génération sonore continue, exploitant deux *buffers* circulaires. Le premier contient les échantillons audio issus de la synthèse vocale incrémentale, qui accompagne la saisie de l'utilisateur. Le second est un *buffer* de "silence", qui contient des échantillons d'amplitude faible (et aléatoire), et permettant une transition lisse entre les différents groupes de mots synthétisés. Par ailleurs, ce dernier *buffer* a, à terme, vocation à accueillir des éléments para-linguistiques de type hésitations (euh) ou bruits de respiration, afin d'améliorer potentiellement le naturel de l'interaction.

Le logiciel, nommé iTTS (pour *incremental Text-to-Speech*), est basé sur l'API audio temps-réel *rtAudio* (API : *Application Programming Interface*). Il s'agit d'une API *opensource* et multi-plateforme, adaptée au traitement temps-réel [1]. Pour garder l'aspect multi-plateforme, nous avons utilisé pour le reste du développement l'API Qt [2].

4.3.2 Architecture client-serveur

Le second prototype développé est basé sur une architecture client-serveur (de type *cloud-computing*), et vise une utilisation sur tablette ou *smartphone*. Dans cette dernière, la synthèse est réalisée sur le serveur et les données d'entrée (paramètres de synthèse, texte saisi par l'utilisateur) et de sortie (échantillons audio de la voix synthétisée) sont diffusées sur le réseau (*streaming*) entre le client et le serveur. Par ailleurs, l'architecture client-serveur permet à plusieurs utilisateurs de "discuter" ensemble, par synthèse incrémentale, à l'aide d'un mode "salon de discussion". Une telle interface présente l'avantage d'être beaucoup plus facile à utiliser : elle ne nécessite pas d'installation ni de mise à jour et peut être utilisée sur tout type de plate-forme à partir d'un navigateur internet. Cette facilité d'utilisation peut notamment nous amener à envisager une utilisation du prototype dans un hôpital ou chez un ergothérapeute. Par ailleurs, le fait de pouvoir l'utiliser sur une tablette ou un smartphone peut permettre une correction automatique des mots mal orthographiés (ou mal conjugués), une saisie semi-automatique ou même une prédiction des mots suivants grâce aux logiciels d'accompagnement de la saisie existants sur ce type d'appareils.

Le schéma de la Figure 4.4 en résume le principe de fonctionnement : chaque client souhaitant se connecter instancie un synthétiseur incrémental, avec ses propres paramètres. Le résultat de la synthèse est accessible à tous les utilisateurs du salon de discussion. Une capture d'écran du prototype web de synthétiseur incrémental est présentée à la Figure 4.5.

1. disponible à l'adresse `https://www.music.mcgill.ca/~gary/rtaudio/`
2. `https://www.qt.io`

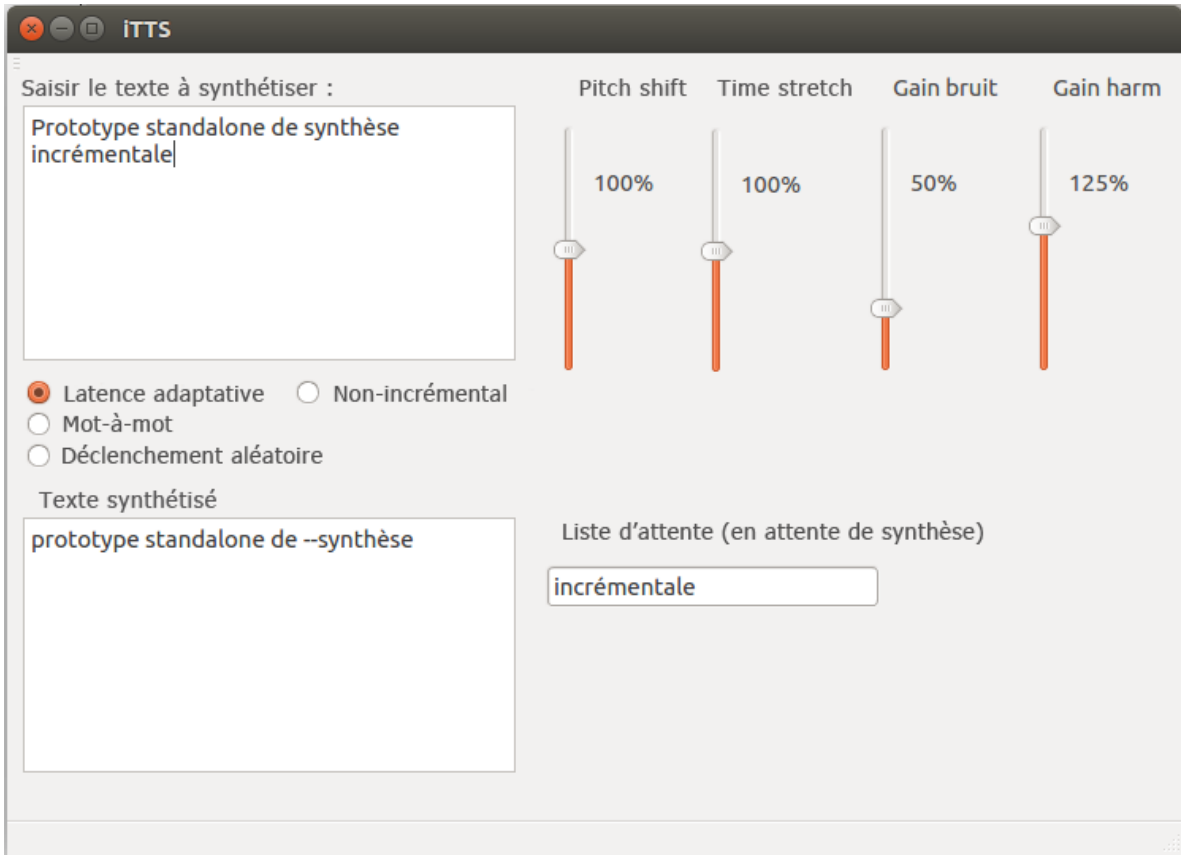


FIGURE 4.3 – Capture d'écran du logiciel iTTS

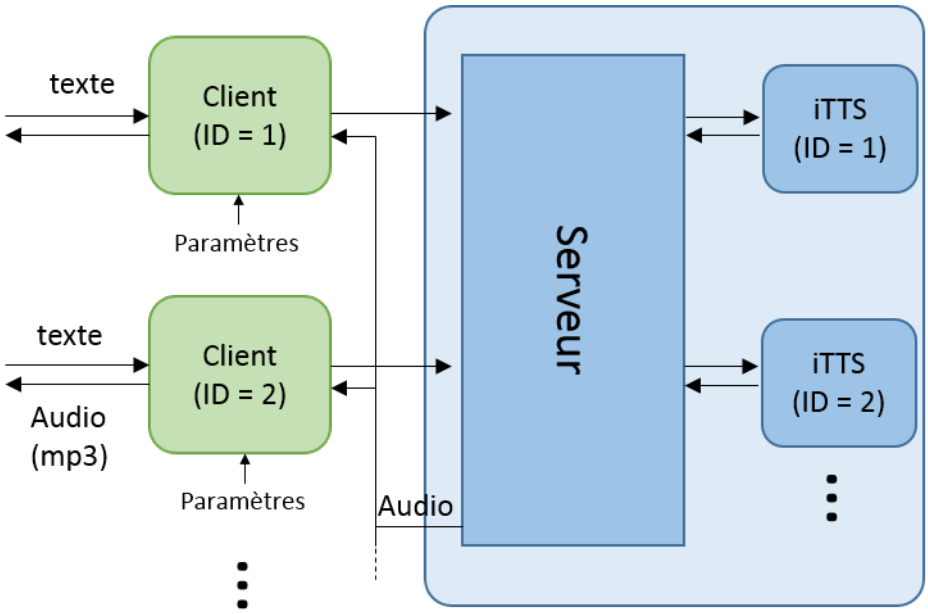


FIGURE 4.4 – Schéma-bloc du prototype client-serveur permettant à plusieurs utilisateurs de communiquer grâce à de la synthèse incrémentale

4.3.3 Adaptation de la synthèse sonore à la vitesse de saisie

Pour la plupart des utilisateurs, la vitesse de saisie du texte (environ 20 mots par minutes (KARAT et al. 1999)) est généralement beaucoup plus faible que la vitesse "d'oralisation" (110 à 140 mots par minutes, 180 mots par minutes pour le corpus d'entraînement des HMM) de la parole de synthèse (et à plus forte raison pour les utilisateurs en situation de handicap). Ce constat est illustré à la figure 4.6a.

Pour limiter le temps entre la synthèse de 2 groupes de mots, nous proposons de ralentir la parole de synthèse afin de l'adapter à la vitesse de saisie. Il s'agit donc d'un problème de dilatation temporelle (*time-stretching*) que nous pouvons résoudre soit au niveau de la génération des durées (qui permet de tenir compte de l'élasticité des états), soit au niveau du vocodeur HNM (c'est ce dernier choix qui a été adopté pour l'implémentation de ce prototype). Le résultat d'une diminution de la vitesse de synthèse est illustré en figure 4.6b. Le facteur de ralentissement est fixé par l'utilisateur (via l'interface graphique).

4.4 Évaluation perceptive du système complet

4.4.1 Méthode d'évaluation

La qualité d'une conversation effectuée à l'aide d'un système de synthèse incrémentale, tel que celui développé dans le cadre de cette thèse, dépend de plusieurs facteurs :

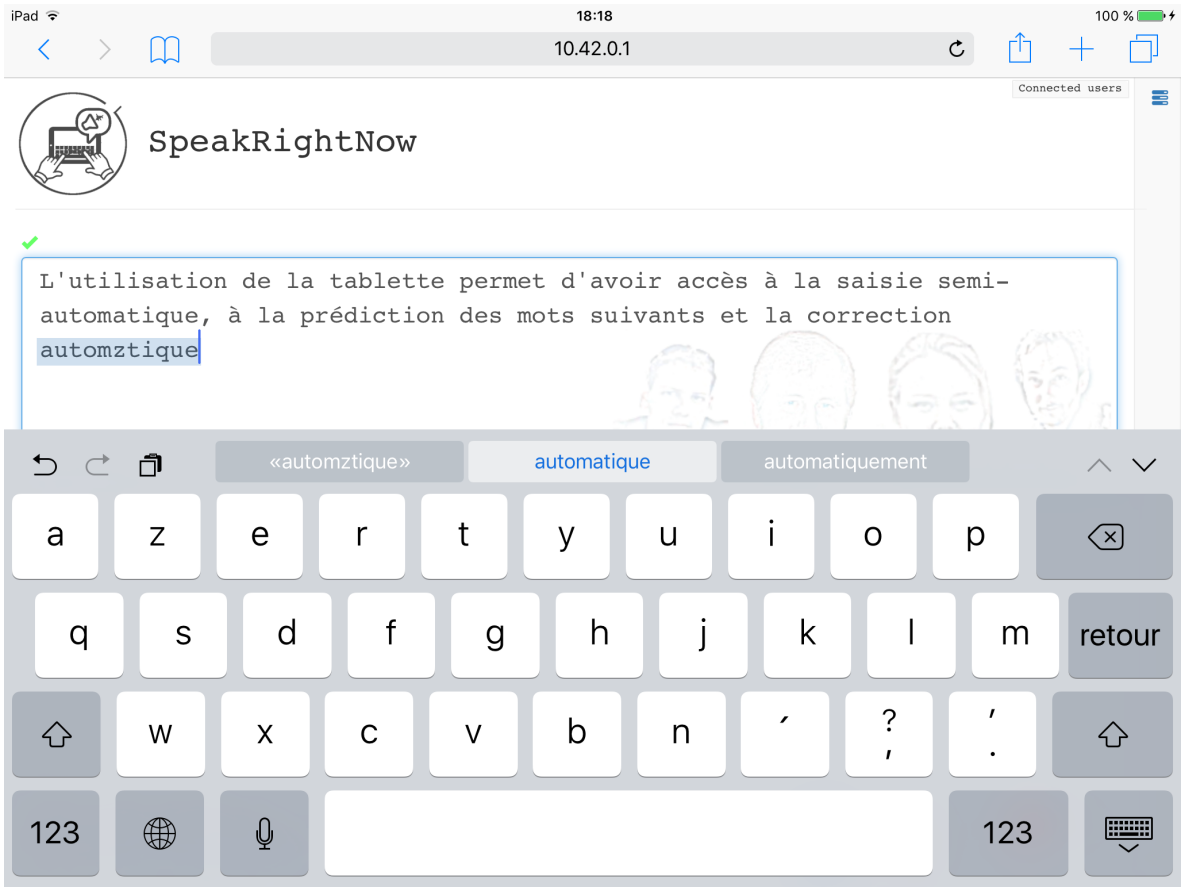
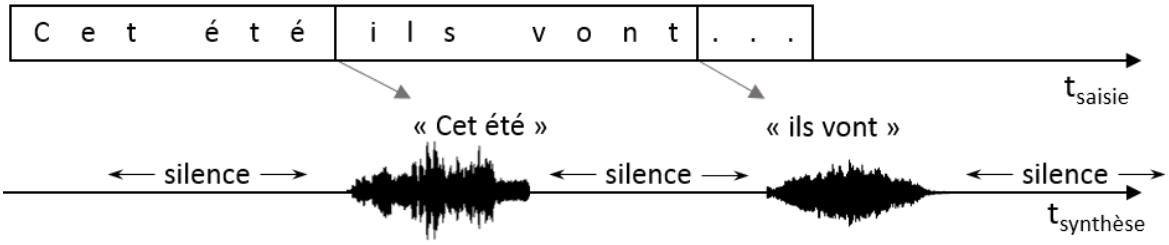
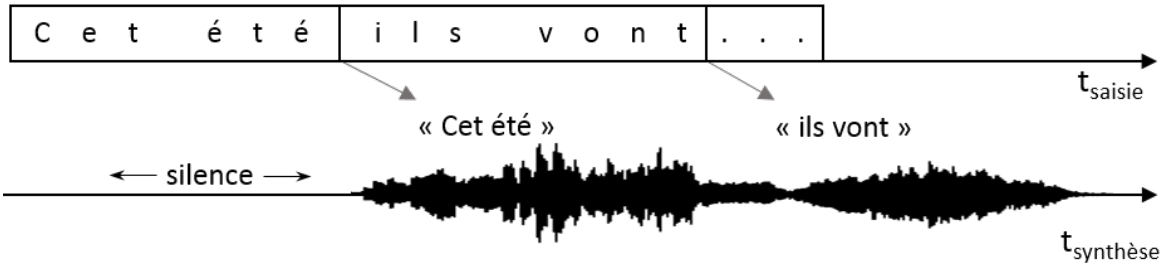


FIGURE 4.5 – Capture d'écran de l'application client-serveur de synthèse incrémentale sur tablette



(a) Vitesse de saisie inférieur à la vitesse de synthèse : Si le segment de parole est synthétisé à vitesse normale, la saisie est plus lente que la synthèse. Cette différence de vitesse conduit à l'apparition de silences entre les segments audio.



(b) Vitesse de saisie comparable à la vitesse de synthèse : si la vitesse de synthèse est diminuée, la parole peut être générée de façon continue (au prix cependant d'une voix ralentie)

FIGURE 4.6 – Illustration des conséquences des différences entre vitesse de saisie du texte et débit audio. Les carrés encadrant les mots servent à représenter les déclenchements de synthèse

— La qualité intrinsèque du système qui dépend donc de l'analyse morpho-syntaxique (ici à latence adaptative) et de la qualité des voix de synthèse par HMM (entraînées ici à l'aide de la stratégie proposée "Joker").
— Des facteurs extrinsèques liés à la manière dont un utilisateur utilise le système (vitesse de frappe, anticipation du texte à écrire, faute de frappe, co-adaptation utilisateur-système, etc.).

Les facteurs extrinsèques sont assez difficiles à contrôler du fait d'une potentielle grande variabilité entre les utilisateurs (en situation de handicap ou non) et des types d'interactions. Aussi, dans cette thèse, nous nous sommes focalisés sur l'évaluation de la qualité intrinsèque du système. À l'aide d'un test perceptif d'écoute, nous avons cherché à évaluer la qualité du couplage des modules de TAL et de synthèse sonore incrémentaux. Cette qualité dépend principalement 1) de la pertinence des regroupements de mots issus de l'analyse morpho-syntaxique à latence adaptative, et 2) de la qualité de la prosodie obtenue lors de leur synthèse.

Pour ce faire, nous demandons à des sujets de comparer simultanément quatre versions d'une même phrase. Ces versions se distinguent par le regroupement des mots qui composent la phrase.

Les phrases d'évaluation sont découpées selon quatre stratégies :
— Stratégie 1 : **Mot-à-mot**. Cette stratégie consiste à déclencher la synthèse à chaque fois qu'un nouveau mot est disponible.
— Stratégie 2 : **Regroupement aléatoire**. Dans cette stratégie, la variable "décision de déclenchement de la synthèse" est remplacée par une valeur binaire aléatoire. La synthèse d'un mot étant toujours conditionnée par la synthèse des mots qui le précèdent, cela permet la construction de groupes de taille aléatoire. Cette stratégie sert, a priori, de condition contrôle.
— Stratégie 3 : **Découpage expert**. Pour réaliser le découpage, nous avons demandé à des experts (humains) de lire la phrase dans sa totalité et de placer des séparateurs entre chaque groupe intonatif.
— Stratégie 4 : **Synthèse incrémentale à latence adaptative**. Il s'agit du découpage s'appuyant sur la stabilité des classes lexicales (méthode proposée).

Les stimuli ainsi découpés sont synthétisés selon la méthode illustrée en figure 4.2, i.e. avec le jeu de modèles HMM "Joker" à taille de contexte droit variable.

Les phrases choisies pour réaliser ce test perceptif sont 14 phrases extraites du corpus de (Combescure 1981) et sont détaillées dans le tableau A.

La durée des silences entre chaque groupe de mots est contrainte de façon à ce que les durées des quatre versions d'une phrases soient toutes égales et que la durée minimale d'un silence entre deux groupes de mots soit de 300ms (durée arbitraire). Au sein d'un même stimulus, tous les silences ont une durée identique.

Le protocole expérimental utilisé pour ce test est identique à celui mis en œuvre au chapitre précédent. Le sujet doit évaluer les 4 versions d'une même phrase, et les ordonner sur une échelle horizontale continue, à l'aide de la même interface, rappelée à la Figure 4.7.

Ce test est réalisé dans un environnement calme avec un casque audio, par 20 sujets de

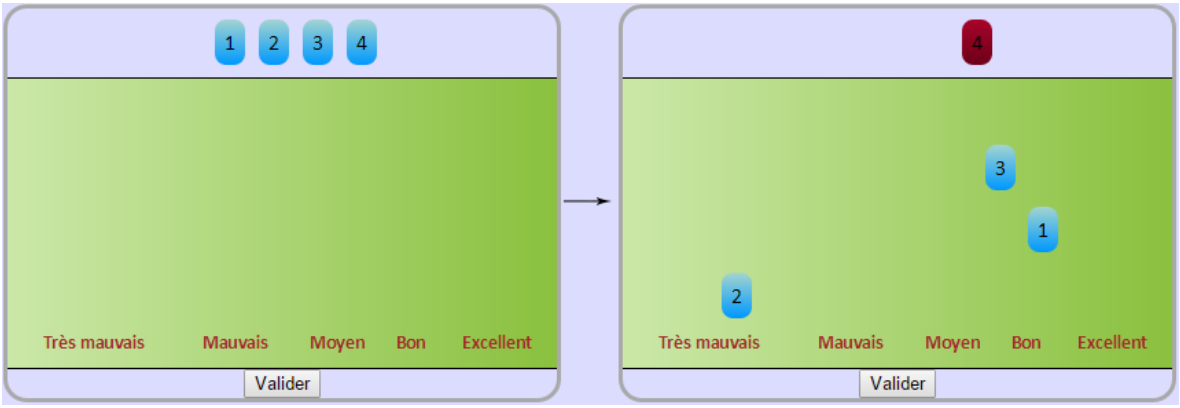


FIGURE 4.7 – Interface utilisée pour l'évaluation perceptive du système de synthèse TTS incrémentale complet. Le sujet doit double-cliquer sur un stimulus pour l'écouter et pouvoir ensuite le placer sur la grille de notation (il est précisé que la position verticale du stimulus sur la grille de notation n'est pas importante). Une fois les quatre stimuli placés sur la grille, ils sont rejoués de celui étant considéré comme pire à celui étant considéré comme meilleur avant de pouvoir passer à la phrase suivante.

langue maternelle française, sans expertise en synthèse de parole ni en phonétique. L'ordre des stimuli rendu aléatoire pour chaque participant et, pour un stimulus donné, l'ordre de présentation des stratégies était également mélangé. De façon similaire au chapitre précédent, la signification statistique des résultats est évaluée à l'aide d'une régression bêta, en considérant la valeur des abscisses comme la variable à expliquer et la stratégie de regroupement comme l'effet fixe explicatif.

4.4.2 Résultats et discussion

Les résultats du test perceptif sont présentés en figure 4.8. Comme nous pouvions nous y attendre, la stratégie de découpage jugée comme étant la plus naturelle par les participants est celle réalisée par des experts humains ayant accès à l'ensemble du contenu sémantique et rythmique de la phrase. De façon surprenante, nous constatons que la stratégie consistant à déclencher la synthèse après chaque saisie de mots – il s'agit de la méthode qui semble la plus intuitive lorsqu'on pense à de la synthèse incrémentale – présente un score de préférence inférieur au regroupement aléatoire qui était pourtant notre condition contrôle. Il semblerait donc qu'un auditeur privilégie un regroupement "prosodique" incohérent avec la syntaxe, plutôt qu'une synthèse mot-à-mot.

Nous notons également que le regroupement basé sur la méthode de latence adaptative arrive en deuxième position après le regroupement expert. Cette méthode a donc été jugée significativement meilleure que le regroupement aléatoire et que la synthèse mot-à-mot. Ce dernier résultat tend à valider la pertinence de l'approche proposée dans cette thèse pour la création d'un système TTS incrémental réalisant un bon compromis entre réactivité du système et qualité de la parole de synthèse.

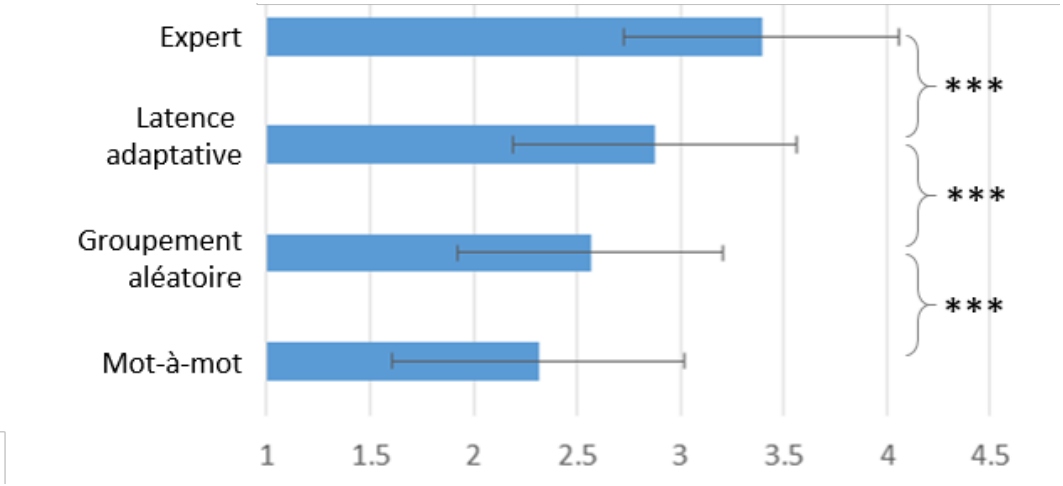


FIGURE 4.8 – Résultats de l'évaluation perceptive de la qualité et du rendu sonore des regroupements de mots (issus de l'analyse morpho-syntaxique à latence adaptative). Sont représentés sur ce graphique la valeur réponse moyenne et l'écart-type pour les 4 stratégies à évaluer. *** signifie une p-value < 0.001 lors des comparaisons multiples.

Le test proposé montre une préférence de la part des sujets pour un regroupement des mots basé sur la stabilité de la classe lexicale plutôt qu'aucun regroupement. Cependant, ce test montre qu'une réflexion sur une autre méthode de regroupement peut être menée afin d'atteindre la qualité du regroupement "expert". Le test proposé pour évaluer la qualité des regroupement ne rend cependant pas compte du temps passé à écrire chaque mot (ou groupe de mots). De futurs travaux devraient donc porter sur une étude en condition réelle d'utilisation des capacités de regroupement et de synthèse du synthétiseur incrémental développé au cours de cette thèse. Ces évaluations pourraient porter sur la pertinence de l'incrémentalité dans le contexte d'une interaction entre deux sujets (avec l'un ou ou les deux participants ayant recours à un synthétiseur incrémental) ou dans la réalisation conjointe d'une tâche spécifique où la synthèse incrémentale constitue un réel avantage.

4.5 Conclusions et perspectives

Nous avons, au cours de ce chapitre, présenté deux versions d'un prototype de synthétiseurs TTS incrémental, basé sur les contributions méthodologiques présentées aux chapitres précédents. Ce prototype pourrait à terme être utilisé par une personne handicapée utilisant la synthèse TTS comme outil de suppléance vocale pour communiquer avec son entourage ou par plusieurs utilisateurs interagissant dans un salon de discussion en ligne (messagerie instantanée). Une première évaluation perceptive a été mise en place pour évaluer la qualité du couplage entre analyse morpho-syntaxique à latence adaptative et modèle de synthèse HMM entraînés à l'aide de la stratégie incrémentale "Joker". Dans le système basé sur ce couplage, la parole de synthèse est délivrée par groupes de mots de taille variable (et ce afin de garantir la stabilité de la classe lexicale). Cette première évaluation perceptive évalue donc

notamment la pertinence de ces regroupements. Les résultats expérimentaux montre que le regroupement obtenu par notre approche est préféré à un regroupement aléatoire ou à une synthèse "mot-à-mot". Ceci valide donc en partie l'approche développée dans cette thèse.

Plusieurs améliorations et évaluations pourront être envisagées par la suite : il serait notamment intéressant de proposer une méthode de regroupement se rapprochant du regroupement "expert" proposé lors de l'expérience perceptive. Cette méthode de regroupement pourrait par exemple reposer sur une estimation incrémentale des syntagmes constituant la phrase à synthétiser telle que proposée par (Beuck, Köhn et Menzel 2013). Il pourrait également être intéressant d'introduire des éléments para-linguistiques (hésitations, respirations, etc.) entre la synthèse de deux groupes de mots, afin d'indiquer à l'interlocuteur que la phrase en cours n'est pas encore finie. Enfin, il serait également intéressant de proposer une évaluation impliquant de véritables utilisateurs, potentiellement en situation de handicap, et utilisant le système dans des conditions écologiques.

Chapitre 5

Conclusions et Perspectives

Bilan de la problématique et des contributions

Au cours de nos travaux, nous nous sommes intéressé à la synthèse de la parole à partir du texte au cours de sa saisie par l'utilisateur, paradigme que nous appelons synthèse *incrémentale* de la parole.

Les systèmes TTS standards sont basés sur un module de traitement automatique de la langue naturelle (TAL) pour effectuer l'analyse linguistique de la phrase à synthétiser et sur un module de synthèse sonore pour transformer les informations linguistiques issues du module de TAL en un signal audio de parole. L'analyse linguistique vise notamment à extraire la structure syntaxique de la phrase à synthétiser, ce qui aide à la phonétisation et à la génération de la prosodie cible. Le paradigme classique utilisé en synthèse TTS est la synthèse de phrases entières (ou de paragraphes). Dans ce cas, les modules de TAL et de synthèse sonore peuvent s'appuyer sur le contexte gauche et droit pour l'analyse de chaque mot. En synthèse incrémentale, l'analyse linguistique et la synthèse sonore ne peuvent se faire qu'à partir des mots déjà saisis, ce qui en limite a priori la précision. Au cours de cette thèse, nous avons cherché à rendre incrémental un système TTS en travaillant d'une part sur le module de traitement automatique de la langue naturelle, au chapitre 2, et d'autre part sur le module de synthèse sonore par HMM, au chapitre 3. Les solutions que nous avons proposées pour rendre ces deux modules incrémentaux nous ont permis de développer un prototype complet de synthèse de parole à partir du texte, qui a été présenté au chapitre 4. Les contributions de ce travail sont brièvement résumées ci-après.

Traitement automatique de la langue naturelle incrémental

Le module de traitement automatique de la langue naturelle (TAL) permet notamment de calculer, à partir d'une séquence de mots, la séquence de phonèmes, le découpage en syllabes, les classes lexicales (analyse morpho-syntaxique) et le découpage en syntagmes (analyse structurelle). Dans ce travail de thèse, nous nous sommes focalisés sur la détermination des classes lexicales (*POS-tagging*) dans le cadre de la synthèse incrémentale.

Nous avons proposé un algorithme visant à déterminer si la classe lexicale inférée uniquement à partir du contexte gauche est susceptible de changer avec l'ajout de mots supplémentaires. Cette estimation est réalisée à partir des classes lexicales (et des probabilités associées) des trois derniers mots tapés. L'algorithme peut alors estimer que la classe lexicale associée au

dernier mot saisi est soit stable - il déclenche sa synthèse - soit instable, c'est-à-dire qu'elle est susceptible de changer avec l'ajout du mot suivant. Dans ce cas, la synthèse est retardée et ce même mot sera réanalysé après l'ajout d'un mot supplémentaire (donc avec un contexte droit d'un mot). Cette méthode d'analyse morpho-syntaxique dite à "latence adaptative" pour la synthèse incrémentale nous permet d'estimer les classes lexicales avec une précision de 92.5% pour une latence moyenne de 1.4 mots (contre 58%, 89% et 97% en cas de latence fixe égale respectivement à 0, 1 ou 2 mots).

Synthèse sonore par HMM incrémentale

Dans ce travail de thèse, nous nous sommes intéressé à l'adaptation d'un module de synthèse sonore par HMM au paradigme de la synthèse incrémentale. Nous avons proposé d'intégrer l'absence possibles de connaissances sur le contexte droit à l'entraînement des HMM. Ceci se caractérise par l'utilisation d'un label "Joker" lorsqu'un descripteur contextuel ne peut pas être calculé. Ces labels ont également été intégrés aux questions permettant de partitionner l'espace acoustique afin de permettre le regroupement d'états partageant une incertitude sur un descripteur associé au contexte droit. La méthode proposée a été comparée à deux méthodes de référence (Baumann 2014 et Astrinaki 2014) à l'aide de tests objectifs et perceptifs. Les résultats obtenus tendent à montrer la pertinence de la méthode proposée (qui fournit de meilleures performances que les méthodes de référence), du moins pour la langue française. Par ailleurs, la qualité de la voix de synthèse mise en œuvre avec la technique proposée pour la synthèse incrémentale est très proche de celle obtenue avec un entraînement classique (pour la synthèse non-incrémentale, c'est-à-dire exploitant les contextes gauche et droit).

Prototype complet

Les techniques d'analyse morpho-syntaxique à latence adaptative et d'entraînement des voix de synthèse par HMM pour la synthèse TTS incrémentale, proposées dans ce travail, ont été couplées et implémentées dans un prototype complet de synthèse TTS incrémentale (en langue française). Une évaluation perceptive de la qualité de ce couplage a été menée afin d'évaluer la pertinence des regroupements de mots imposés par l'analyse morpho-syntaxique à latence adaptative. Cette évaluation montre que l'approche proposée est préférée à une synthèse mot-à-mot et se rapproche d'un groupement "expert" (réalisé par des transcripteurs humains). L'ensemble de ces résultats, à savoir la pertinence des regroupements et la qualité de la voix de synthèse incrémentale nous permettent de proposer un système qui satisfait le compromis entre qualité et réactivé évoqué en introduction, et dont la recherche était l'objectif principal de ce travail de thèse.

Ce travail est un premier pas vers la synthèse TTS incrémentale et de multiples améliorations sont envisageables. Nous résumons dans la section suivante quelques perspectives se dégageant de ce travail.

Perspectives

Le module d'analyse morpho-syntaxique que nous utilisons actuellement repose sur les modèles *n-gram* pour l'estimation de la classe lexicale. Ces modèles exploitent la probabilité pour un mot d'appartenir à une classe lexicale, sachant les n classes lexicales précédentes. De futures pistes de réflexions pour réaliser l'étiquetage morpho-syntaxique d'un texte en cours de saisie pourraient faire usage de modèles prédictifs capables de modéliser des dépendances linguistiques à plus long terme, comme les LSTM-RNN (*Long Short-Term Memory Recurrent Neural Network* (Hochreiter et Schmidhuber 1997).

Par ailleurs, l'ensemble de l'analyse linguistique incrémentale pourrait bénéficier de l'approche récente de prolongements de mots (*word embedding*) `Word2Vec` (Mikolov et al. 2013). Cette dernière vise à extraire des descripteurs de haut-niveau qui encodent la sémantique d'un texte. En introduisant de l'information rendant compte de l'ordre des mots au sein d'un énoncé, (Ling et al. 2015) extraient depuis le texte des descripteurs de haut-niveau rendant compte de la structure syntaxique de la phrase et pouvant être utilisés pour réaliser l'analyse morpho-syntaxique d'un énoncé.

Afin d'enrichir la liste des descripteurs contextuels utilisables par le module de synthèse sonore incrémentale, nous souhaitons ajouter les descripteurs se rapportant à l'analyse structurelle (fournissant des informations sur syntagmes et permettant de placer des marqueurs délimitant les groupes intonatifs). Ces informations seraient d'autant plus intéressantes qu'elles permettraient également de fournir un schéma de regroupement des mots lors de la synthèse (plutôt que de se baser sur l'analyse morpho-syntaxique ; à condition que les classes lexicales soit correctement estimées). La question de l'analyse structurelle incrémentale a déjà été abordée notamment par (Mori, Matsubara et Inagaki 2001) ou, plus récemment par (Beuck, Köhn et Menzel 2013).

En ce qui concerne le module de synthèse sonore, nous envisageons deux axes d'amélioration principaux. Il semble que l'approche par HMM laisse progressivement sa place au réseaux de neurones profonds et récurrents (Zen 2015 ; Zen et Sak 2015). Aussi, il pourrait être intéressant d'adapter la méthode de construction de la voix de synthèse incrémentale proposée (méthode "Joker") à cette approche. Cela consisterait à entraîner un réseau de neurones profond en intégrant dans les descripteurs contextuels un marqueur explicite d'une incertitude sur le contexte droit.

Le second axe d'amélioration de la qualité de la synthèse incrémentale concerne le corpus d'apprentissage. Actuellement, le corpus utilisé est extrait d'un livre audio : il s'agit donc de parole lue dont la prosodie peut largement différer de celle d'une parole spontanée. Or, lorsque la synthèse incrémentale est utilisée pour de la suppléance vocale, nous pouvons faire l'hypothèse qu'une parole "spontanée" sera privilégiée par l'utilisateur. Aussi, il serait intéressant d'entraîner la voix de synthèse incrémentale (à l'aide de la méthode proposée "Joker") sur un corpus de parole spontanée. Une situation intermédiaire serait l'enregistrement d'un corpus dans lequel le locuteur dispose d'une connaissance limitée sur le contexte droit du texte qu'il doit prononcer (ce dernier se dévoilant au fur et à mesure de la lecture).

Enfin, le prototype complet, couplant la méthode d'analyse morpho-syntaxique à latence adaptative et la technique d'entraînement d'une voix de synthèse incrémentale a fait l'objet d'une évaluation "hors-ligne" (test perceptif visant à évaluer la qualité et la pertinence des groupes de mots synthétisés au fur et à mesure de la saisie). Une évaluation "en ligne", c'est-à-dire dans le cadre d'une véritable interaction conversationnelle est nécessaire. Cette évaluation permettrait de réaliser diverses mesures objectives lors de la réalisation de tâches (à deux participants) dans l'optique de quantifier le bénéfice de l'approche incrémentale (par rapport à une synthèse mot-à-mot et à un synthétiseur non-incrémental phrase-à-phrase).

Enfin, la synthèse incrémentale pourrait être utilisée dans d'autres contextes, comme par exemple les systèmes de traduction automatique en temps réel (Bangalore et al. 2012) ou les systèmes de conversion de signaux physiologiques en parole tels que les interfaces cerveaux-machine (Bocquelet 2017) ou les interfaces de communication en parole silencieuse (Hueber 2009).

Bibliographie

Allen, J, S Hunnicut et D Klatt (1987). *From Text to Speech, the MITTALK system.* Cambrige University Press, USA (cf. p. 20).

Astrinaki, M. et al. (2012). "Reactive and continuous control of HMM-based speech synthesis". In : *Proceedings of Spoken Language Technology Workshop (SLT), IEEE.* Miami, FL, USA, p. 252–257 (cf. p. 77).

Astrinaki, Maria (2014). "Performative Statistical Parametric Speech Synthesis Applied to Interactive Designs". Thèse de doct. Mons, Belgique : Numédiart (cf. p. 11, 68, 76, 104).

Bailly, Gérard et Mamoun Alissali (1992). "Compost : un serveur de synthèse de parole multilingue". In : *Traitement du Signal* 9.4, p. 359–366 (cf. p. 18, 33, 73).

Bailly, Gérard et Ian Gorisch (2006). "Generating German intonation with a trainable prosodic model". In : *Proceedings of Interspeech.* Pittsburgh, PA, USA, p. 2366–2369 (cf. p. 84).

Bailly, Gérard et Cécilia Gouvernayre (2012). "Pauses and respiratory markers of the structure of book reading". In : *Proceedings of Interspeech.* Portland, OR, USA, p. 2218–2221 (cf. p. 73).

Bangalore, Srinivas et al. (2012). "Real-time incremental speech-to-speech translation of dialogs". In : *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies.* Montréal, Canada : Association for Computational Linguistics, p. 437–445 (cf. p. 106).

Baumann, T. (2014). "Decision tree usage for incremental parametric speech synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* Florence, Italy, p. 3819–3823 (cf. p. 11, 69, 74, 75, 79, 87, 104).

Baumann, Timo (2013). "Incremental Spoken Dialogue Processing : Architecture and Lower-level Components". Thèse de doct. Universität Bielefeld, Germany (cf. p. 5).

Baumann, Timo et David Schlangen (2012a). "Evaluating Prosodic Processing for Incremental Speech Synthesis". In : *Proceedings of Interspeech.* Portland, OR, USA, p. 438–441 (cf. p. 76).

— (2012b). "INPRO_iSS : A component for just-in-time incremental speech synthesis". In : *Proceedings of the ACL 2012 System Demonstrations.* Jeju Island, Korea : Association for Computational Linguistics, p. 103–108 (cf. p. 11).

— (2012c). "The INPROTK 2012 release". In : *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies.* Montréal, Canada, p. 29–32 (cf. p. 11, 69).

Beuck, Niels, Arne Köhn et Wolfgang Menzel (2011). "Decision Strategies in Incremental PoS Tagging". In : *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA).* Riga, Latvia, p. 26–33 (cf. p. 12, 22).

— (2013). "Predictive incremental parsing and its evaluation". In : *Computational Dependency Theory.* Vol. 258. Kim Gerdes, Eva Hajičová, Leo Wanner, p. 186 (cf. p. 101, 105).

Bilmes, Jeff A. (1998). "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models". In : *International Computer Science Institute* 4.510, p. 126 (cf. p. 55).

Blache, Philippe et Stéphane Rauzy (2007). "Le module de reformulation iconique de la Plateforme de Communication Alternative". In : *Actes de la 14ème conférence sur le Traitement Automatique des Langues Naturelles (TALN)*. Toulouse, France, p. 519–528 (cf. p. 6).

Bloit, Julien et Xavier Rodet (2008). "Short-time Viterbi for online HMM decoding : Evaluation on a real-time phone recognition task". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Las Vegas, NV, USA, p. 2121–2124 (cf. p. 26).

Bocquelet, Florent (2017). "Toward a Brain-Computer Interface for speech rehabilitation". Thèse de doct. Grenoble, France : Université Grenoble-Alpes (cf. p. 106).

Boite, René et al. (2000). *Traitement de la parole*. Presses Polytechniques et Universitaires Romandes (cf. p. 8, 16, 43).

Braga, Daniela, Luís Coelho et Fernando Gil Vianna Resende (2006). "A rule-based grapheme-to-phone converter for TTS systems in European Portuguese". In : *Telecommunications Symposium, 2006 International*. IEEE, p. 328–333 (cf. p. 20).

Brants, Thorsten (2000). "TnT : a statistical part-of-speech tagger". In : *Proceedings of the sixth conference on Applied natural language processing*. Seattle, WA, USA, p. 224–231 (cf. p. 18).

Breiman, Leo et al. (1984). *Classification and regression trees*. CRC press (cf. p. 30, 34).

Büring, Daniel (2013). "Syntax, information structure and prosody". In : *The Cambridge Handbook of Generative Syntax*. Marcel den Dikken, p. 860–895 (cf. p. 56).

Buschmeier, Hendrik et Stefan Kopp (2011). "Towards Conversational Agents That Attend to and Adapt to Communicative User Feedback". In : *Proceedings of the 11th International Conference on Intelligent Virtual Agents*. Reykjavik, Iceland, p. 169–182 (cf. p. 11).

Cadic, Didier (2011). "Optimised voice creation for unit-selection synthesis". Theses. Université Paris Sud - Paris XI (cf. p. 12).

Che, Hao, Jianhua Tao et Shifeng Pan (2012). "Letter-to-sound conversion using coupled Hidden Markov Models for lexicon compression". In : *Speech Database and Assessments (Oriental COCOSDA), 2012 International Conference on*. IEEE. Macau, China, p. 141–144 (cf. p. 21).

Chen, Stanley F et Joshua Goodman (1999). "An empirical study of smoothing techniques for language modeling". In : *Computer Speech & Language* 13.4, p. 359 –394 (cf. p. 24).

Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In : *Journal of Machine Learning Research* 12.Aug, p. 2493–2537 (cf. p. 18).

Combescure, Pierre (1981). "listes de dix phrases phonétiquement équilibrées". In : *Revue d'acoustique* 56 (cf. p. 98).

Constant, Matthieu et al. (2011). "Intégrer des connaissances linguistiques dans un CRF : application à l'apprentissage d'un segmenteur-étiqueteur du français". In : *TALN*. Vol. 1. Montpellier, France, p. 321 (cf. p. 18).

d'Alessandro, Christophe (2001). "33 ans de synthèse de la parole à partir du texte : une promenade sonore (1968-2001)". In : *Traitement Automatique des Langues* 42.1, p. 1–29 (cf. p. 43, 44).

D'Alessandro, Nicolas et Thierry Dutoit (2007). "HandSketch bi-manual controller : investigation on expressive control issues of an augmented tablet". In : *Proceedings of the 7th international conference on New interfaces for musical expression*. New York, NY, USA, p. 78–81 (cf. p. 11).

De Cheveigné, Alain et Hideki Kawahara (2002). "YIN, a fundamental frequency estimator for speech and music". In : *The Journal of the Acoustical Society of America* 111.4, p. 1917–1930 (cf. p. 50).

Di Cristo, Albert (2000). "Interpréter la prosodie". In : *Actes des XXIIIèmes Journées d'Etude sur la Parole*. Aussois, p. 13–29 (cf. p. 2).

Drugman, T. et al. (2009). "Using a pitch-synchronous residual codebook for hybrid HMM/frame selection speech synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Taipei, Taiwan, p. 3793–3796 (cf. p. 50).

Drugman, Thomas, Geoffrey Wilfart et Thierry Dutoit (2009). "A deterministic plus stochastic model of the residual signal for improved parametric speech synthesis." In : *Proceedings of Interspeech*. Brighton, UK, p. 1779–1782 (cf. p. 50).

Dutoit, Thierry et al. (2011). "pHTS for Max/MSP : A Streaming Architecture for Statistical Parametric Speech Synthesis". In : *QPSR of the numediart research program* 4.1, p. 7–11 (cf. p. 11).

Edlund, Jens (2008). "Incremental speech synthesis". In : *Proceedings of Swedish Language Technology Conference*. Stockholm, Sweden, p. 53–54 (cf. p. 11).

Fan, Y. et al. (2015). "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Brisbane, Australia, p. 4475–4479 (cf. p. 45).

— (2016). "Speaker and language factorization in DNN-based TTS synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Shanghai, China, p. 5540–5544 (cf. p. 45).

Ferrari, Silvia et Francisco Cribari-Neto (2004). "Beta regression for modelling rates and proportions". In : *Journal of Applied Statistics* 31.7, p. 799–815 (cf. p. 86).

Feugère, Lionel (2013). "Gestural control of singing voice synthesis by rules and musical applications". Theses. Université Pierre et Marie Curie - Paris VI (cf. p. 11).

Forney, G David (1973). "The viterbi algorithm". In : *Proceedings of the IEEE* 61.3, p. 268–278 (cf. p. 25).

Giménez, Jesús et Lluis Marquez (2004). "SVMTool : A general POS tagger generator based on Support Vector Machines". In : *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Lisbon, Portugal (cf. p. 18).

Gini, Corrado (1912). "Variabilità e mutabilità". In : *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome : Libreria Eredi Virgilio Veschi* 1 (cf. p. 34).

Greene, Barbara B et Gerald M Rubin (1971). *Automated grammatical tagging of English.* Department of Linguistics, Brown University (cf. p. 18).

Guéguin, Marie (2006). "Evaluation objective de la qualité vocale en contexte de conversation". Thèse de doct. Rennes, France : Université Rennes 1 (cf. p. 7).

Hahn, Stefan et al. (2013). "Improving LVCSR with hidden conditional random fields for grapheme-to-phoneme conversion." In : *Proceedings of Interspeech.* Lyon, France, p. 495–499 (cf. p. 21).

Hochreiter, Sepp et Jürgen Schmidhuber (1997). "Long short-term memory". In : *Neural computation* 9.8, p. 1735–1780 (cf. p. 105).

Hothorn, Torsten, Frank Bretz et Peter Westfall (2008). "Simultaneous inference in general parametric models". In : *Biometrical journal* 50.3, p. 346–363 (cf. p. 86).

HTS (2000). *The HTS toolkit* (cf. p. 11, 45, 66, 67).

Hueber, Thomas (2009). "Reconstitution de la parole par imagerie ultrasonore et vidéo de l'appareil vocal : vers une communication parlée silencieuse". Thèse de doct. Paris, France : Université Pierre et Marie Curie (cf. p. 50, 51, 54, 106).

Hunt, Andrew J et Alan W Black (1996). "Unit selection in a concatenative speech synthesis system using a large speech database". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* Atlanta, GA, USA, p. 373–376 (cf. p. 45).

Imai, Satoshi (1983). "Cepstral analysis synthesis on the mel frequency scale". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* Vol. 8, p. 93–96 (cf. p. 48, 49).

Indurkhya, Nitin, Fred J Damerau et David D Palmer (2010). "Text Preprocessing". In : *Handbook of Natural Language Processing, Second Edition.* Chapman et Hall/CRC, p. 9–30 (cf. p. 16).

Jelinek, Frederick et Ciprian Chelba (1999). "Putting language into language modeling". In : *Proceedings of Eurospeech.* Budapest, Hungary (cf. p. 18).

Jiampojamarn, Sittichai et Grzegorz Kondrak (2010). "Letter-phoneme alignment : An exploration". In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics. Uppsala, Sweden, p. 780–788 (cf. p. 21).

Karaali, Orhan, Gerald Corrigan et Ira Gerson (1996). "Speech Synthesis with Neural Networks". In : *World Congress on Neural Networks : International Neural Network Society 1996 Annual Meeting.* San Diego, California, USA : Psychology Press, p. 45 (cf. p. 45).

Karat, Clare-Marie et al. (1999). "Patterns of Entry and Correction in Large Vocabulary Continuous Speech Recognition Systems". In : *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Pittsburgh, PA, USA : ACM, p. 568–575 (cf. p. 95).

Klatt, Dennis H (1987). "Review of text-to-speech conversion for English". In : *The Journal of the Acoustical Society of America* 82.3, p. 737–793 (cf. p. 44).

Kominek, John et Alan W Black (2004). "The CMU Arctic speech databases". In : *Proceedings of Fifth ISCA ITRW on Speech Synthesis (SSW5).* Pittsburg, PA, USA : ed. by Alan W. Black et Kevin Lenzo, p. 223–224 (cf. p. 21).

Kubichek, R.F. (1993). "Mel-cepstral distance measure for objective speech quality assessment". In : *Proceedings of Communications, Computers and Signal Processing, IEEE Pacific Rim Conference on.* Vol. 1. Victoria, British Columbia, Canada, p. 125–128 (cf. p. 77).

Lanchantin, P., G. Degottex et X. Rodet (2010). "A HMM-based speech synthesis system using a new glottal source and vocal-tract separation method". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* Dallas, TX, USA, p. 4630–4633 (cf. p. 50).

Lavergne, Thomas, Olivier Cappé et François Yvon (2010). "Practical Very Large Scale CRFs". In : *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Uppsala, Sweden, p. 504–513 (cf. p. 18).

Le Beux, Sylvain (2009). "Gestural control of prosody and voice quality". Theses. Université Paris Sud - Paris XI (cf. p. 11).

Le Maguer, Sebastien (2013). "Evaluation experimentale d'un systeme statistique de de la parole, HTS, pour la langue francaise." Thèse de doct. Rennes, France : Université de Rennes 1 (cf. p. 56, 68, 78).

Lesk, Michael E et Eric Schmidt (1975). *Lex : A lexical analyzer generator.* Bell Laboratories Murray Hill, NJ (cf. p. 17).

Levinson, S. E., J. P. Olive et J. S. Tschirgi (1993). "Speech synthesis in telecommunications". In : *IEEE Communications Magazine* 31.11, p. 46–53 (cf. p. 20).

Lewis, David D et al. (2004). "Rcv1 : A new benchmark collection for text categorization research". In : *Journal of machine learning research* 5.Apr, p. 361–397 (cf. p. 19).

Ling, Wang et al. (2015). "Two/Too Simple Adaptations of Word2Vec for Syntax Problems". In : *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies.* Denver, Colorado : Association for Computational Linguistics, p. 1299–1304 (cf. p. 105).

Maia, Ranniery et al. (2007). "An excitation model for HMM-based speech synthesis based on residual modeling". In : *Proceedings of the Sixth ISCA Workshop on Speech Synthesis.* Bonn, Germany, p. 131–136 (cf. p. 50).

Mallat, Stéphane (1999). *A wavelet tour of signal processing.* Academic press (cf. p. 59).

Marcus, Mitchell P, Mary Ann Marcinkiewicz et Beatrice Santorini (1993). "Building a large annotated corpus of English : The Penn Treebank". In : *Computational linguistics* 19.2, p. 313–330 (cf. p. 18, 23).

Masuko, Takashi et al. (1997). "Voice characteristics conversion for HMM-based speech synthesis system". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).* Vol. 3, p. 1611–1614 (cf. p. 45).

McTear, Michael F (2004). *Spoken dialogue technology : toward the conversational user interface.* Springer Science & Business Media (cf. p. 20).

Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In : *Advances in Neural Information Processing Systems.* Lake Tahoe, NV, USA, p. 3111–3119 (cf. p. 105).

Mitton, Roger (1992). *A computer-usable dictionary file based on the Oxford Advanced Learner's Dictionary of Current English.* Oxford Text Archive (cf. p. 21).

Mori, Daisuke, Shigeki Matsubara et Yasuyoshi Inagaki (2001). "Incremental parsing for interactive natural language interface". In : *IEEE International Conference on Systems, Man, and Cybernetics*. Vol. 5. IEEE, p. 2880–2885 (cf. p. 12, 105).

Nicol, GT (1993). *Flex - The Lexical Scanner Generator*. Free Software Foundation, Cambridge (cf. p. 17).

Olaszy, Gábor, G Gordos et Géza Németh (1992). "The MULTIVOX multilingual text-to-speech converter". In : *Talking machines : Theories, Models and Applications, Elsevier*, p. 385–411 (cf. p. 44).

Pagel, Vincent, Kevin Lenzo et Alan Black (1998). "Letter to sound rules for accented lexicon compression". In : *5th International Conference on Spoken Language Processing*. Sydney, Australia (cf. p. 21, 33).

Peretz, Isabelle et Krista L Hyde (2003). "What is specific to music processing ? Insights from congenital amusia". In : *Trends in cognitive sciences* 7.8, p. 362–367 (cf. p. 77).

Perrotin, Olivier (2015). "Singing with hands : chironomic interfaces for digital musical instruments". Theses. Université Paris Sud - Paris XI (cf. p. 12).

Pfitzinger, Hartmut R. (1998). "Local Speech Rate As A Combination Of Syllable And Phone Rate". In : *Proceedings of International Conference on Spoken Language Processing (ICSLP)*. Sydney, Australia, p. 1087–1090 (cf. p. 84).

Pouget, Maël et al. (2015). "HMM Training Strategy for incremental speech synthesis". In : *Proceedings of Interspeech*. Dresden, Germany, p. 1201–1205 (cf. p. 13).

Pouget, Maël et al. (2016). "Adaptive Latency for Part-of-Speech Tagging in Incremental Text-to-Speech Synthesis". In : *Proceedings of Interspeech*. San Francisco, CA, United States, p. 2846 –2850 (cf. p. 13).

Raitio, T. et al. (2011). "Utilizing glottal source pulse library for generating improved excitation signal for HMM-based speech synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Prague, Czech Republic, p. 4564–4567 (cf. p. 50).

Rao, Kanishka et al. (2015). "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Brisbane, Australia, p. 4225–4229 (cf. p. 21).

Ribeiro, Manuel Sam, Junichi Yamagishi et Robert A J Clark (2015). "A perceptual investigation of wavelet-based decomposition of f0 for text-to-speech synthesis". In : *Proceedings of Interspeech*. Dresden, Germany, p. 1586–1590 (cf. p. 60).

Richards, D. L. (1973). *Telecommunication by Speech : The Transmission Performance of Telephone Networks*. Butterworths. London (cf. p. 6).

Rosé, Carolyn P, Antonio Roque et Dumisizwe Bhembe (2002). "An efficient incremental architecture for robust interpretation". In : *Proceedings of the second international conference on Human Language Technology Research*. San Diego, California : Morgan Kaufmann Publishers Inc., p. 307–312 (cf. p. 12).

Rossi, Mario et al. (1981). *L'intonation : de l'acoustique à la sémantique*. Klincksieck. Paris (cf. p. 12).

Sagisaka, Yoshinori (1988). "Speech synthesis by rule using an optimal selection of non-uniform synthesis units". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, p. 679–682 (cf. p. 44).

Santos, Cícero Nogueira dos et Bianca Zadrozny (2014). "Learning Character-level Representations for Part-of-Speech Tagging." In : *Proceedings of the 31st International Conference on Machine Learning*. Beijing, China, p. 1818–1826 (cf. p. 18).

Schröder, Marc et Jürgen Trouvain (2003). "The German text-to-speech synthesis system MARY : A tool for research, development and teaching". In : *International Journal of Speech Technology* 6.4, p. 365–377 (cf. p. 18, 69).

Seward, Alexander (2003). "Low-latency incremental speech transcription in the synface project." In : *Proceedings of Interspeech*, p. 1141–1144 (cf. p. 26).

Shinoda, Koichi et Takao Watanabe (2000). "MDL-based context-dependent subword modeling for speech recognition". In : *The Journal of the Acoustical Society of Japan* 21.2, p. 79–86 (cf. p. 63).

Sokolovska, N. et al. (2010). "Efficient Learning of Sparse Conditional Random Fields for Supervised Sequence Labeling". In : *IEEE Journal of Selected Topics in Signal Processing* 4.6, p. 953–964 (cf. p. 18).

Stylianou, Yannis (1996). "Harmonic plus Noise Models for Speech combined with Statistical Methods, for Speech and Speaker Modification". Thèse de doct. Paris, France : Ph.D. thesis, Ecole Nationale superieure des télécommunications (cf. p. 47, 50).

Sun, Xu (2014). "Structure regularization for structured prediction". In : *Advances in Neural Information Processing Systems*. Montréal, Canada, p. 2402–2410 (cf. p. 18).

Sundermeyer, Martin et al. (2011). "The RWTH 2010 quaero ASR evaluation system for English, French, and German". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Prague, Czech Republic, p. 2212–2215 (cf. p. 21).

Suni, Antti et al. (2013). "Wavelets for intonation modeling in HMM speech synthesis". In : *8th ISCA Speech Synthesis Workshop*. barcelona, Spain, p. 285–290 (cf. p. 59).

Tamura, Masatsune et al. (2001). "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 2, p. 805–808 (cf. p. 45).

Taylor, Paul et Alan W Black (1999). "Speech synthesis by phonological structure matching". In : *Proceedings of Eurospeech*. Budapest, Hungary, p. 623–626 (cf. p. 44).

Taylor, Paul, Alan W Black et Richard Caley (1998). "The Architecture of the Festival Speech Synthesis System". In : *Proceedings of the 3rd ESCA/COCOSDA Workshop on Speech Synthesis*. Australia, p. 147–151 (cf. p. 18).

Tokuda, K. et al. (1999). "Hidden Markov models based on multi-space probability distribution for pitch pattern modeling". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. Phoenix, Arizona, USA, 229–232 vol.1 (cf. p. 45).

Tokuda, K. et al. (2000). "Speech parameter generation algorithms for HMM-based speech synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 3. Istanbul, Turkey, p. 1315–1318 (cf. p. 63, 65, 68).

Tokuda, K. et al. (2013). "Speech Synthesis Based on Hidden Markov Models". In : *Proceedings of the IEEE* 101.5, p. 1234–1252 (cf. p. 45, 46).

Tokuda, Keiichi, Takashi Masuko et Noboru Miyazaki (2002). "Multi-Space Probability Distribution HMM". In : *IEICE TRANSACTIONS on Information and Systems* E85.3, p. 455–464 (cf. p. 59).

Toma, Stefan-Adrian et Doru-Petru Munteanu (2009). "Rule-based automatic phonetic transcription for the Romanian language". In : *2009 Computation World : Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*. IEEE, p. 682–686 (cf. p. 20).

Toma, Ştefan-Adrian et al. (2013). "On letter to sound conversion for Romanian : A comparison of five algorithms". In : *7th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE. Cluj-Napoca, Romania, p. 1–6 (cf. p. 20).

Toutanova, Kristina et al. (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network". In : *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Edmonton, Canada, p. 173–180 (cf. p. 18).

Tsai, Chung-Yao et al. (2014). "Hierarchical prosody modeling of English speech and its application to TTS". In : *Proceedings of Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA), 2014 17th Oriental Chapter of the International Committee for the*. Pukhet, Thaïlande, p. 1–6 (cf. p. 56).

Vaissière, Jacqueline (2011). *La phonétique seconde édition révisée*. Presses Universitaires de France (cf. p. 8, 48).

Wang, Dong et Simon King (2011). "Letter-to-sound pronunciation prediction using conditional random fields". In : *IEEE Signal Processing Letters* 18.2, p. 122–125 (cf. p. 21).

Yoshimura, Takayoshi et al. (1998). "Duration modeling for HMM-based speech synthesis." In : *proceedings of ICSLP*. Sydney, Australia, p. 29–31 (cf. p. 58).

— (2000). "Speaker interpolation for HMM-based speech synthesis system." In : *Journal of the Acoustical Society of Japan (E)* 21.4, p. 199–206 (cf. p. 45).

— (2001). "Mixed excitation for HMM-based speech synthesis." In : *Proceedings of Eurospeech*. Aalborg, Denmark, p. 2263–2266 (cf. p. 50).

Young, S. J., J. J. Odell et P. C. Woodland (1994). "Tree-based State Tying for High Accuracy Acoustic Modelling". In : *Proceedings of the Workshop on Human Language Technology*. HLT '94. Stroudsburg, PA, USA : Association for Computational Linguistics, p. 307–312 (cf. p. 60).

Yu, Shun-Zheng (2010). "Hidden semi-Markov models". In : *Artificial Intelligence* 174.2, p. 215 –243 (cf. p. 58).

Yvon, François (2010). *Une petite introduction au Traitement Automatique des Langues Naturelles* (cf. p. 17).

Zen, Heiga (2015). "Acoustic Modeling in Statistical Parametric Speech Synthesis - From HMM to LSTM-RNN". In : *Proceedings of Workshop on Machine Learning in Spoken Language Processing (MLSLP)*. Aizu-Wakamatsu city, Fukushima, Japan, Invited paper (cf. p. 105).

Zen, Heiga et Haşim Sak (2015). "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis". In : *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brisbane, Australia, p. 4470–4474 (cf. p. 45, 87, 105).

Zen, Heiga, Andrew Senior et Mike Schuster (2013). "Statistical parametric speech synthesis using deep neural networks". In : *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver, BC, Canada, p. 7962–7966 (cf. p. 45, 87).

Annexe A

Phrases du corpus de test pour l'évaluation subjective du regroupement de mots

Ci-après sont présentées les phrases utilisées pour l'évaluation de la qualité du regroupement basé sur la stabilité des classes lexicales présentée au chapitre 4. Pour chacune des phrases, sont présentés successivement les découpages

Un	loup	s'est	jeté	immédiatement	sur	la	petite	chèvre.
Un	loup	s'est	jeté	immédiatement	sur la petite chèvre.			
Un loup		s'est jeté		immédiatement	sur la petite chèvre.			
Un loup		s'est	jeté	immédiatement	sur la petite chèvre.			

Le	courrier	du	jour	arrive	en	retard	en	ce moment.
Le	courrier	du jour arrive			en retard		en ce moment.	
Le courrier du jour				arrive en retard			en ce moment	
Le courrier		du jour		arrive	en retard en		ce moment.	

Le	ciel	est	tout	noir,	il	va	tomber	des cordes.
Le	ciel est		tout	noir, il va			tomber des cordes.	
Le ciel		est tout noir			il va tomber			des cordes.
Le ciel		est	tout	noir	il	va tomber des cordes.		

Ces	légendes	me	rappellent	les	temps	anciens.
Ces	légendes me		rappellent	les temps anciens.		
Ces légendes		me rappellent		les temps anciens.		
Ces légendes		me rappellent		les temps		anciens.

Des	gens	se	sont	levés	dans	les	tribunes.
Des	gens se		sont	levés		dans les tribunes.	
Des gens		se sont levés			dans les tribunes		
Des	gens	se sont		levés	dans	les tribunes.	

Souvent	je	m'accoude	au	muret	de	ce	pont.
Souvent	je	m'accoude au		muret	de ce pont.		
Souvent	je m'accoude			au muret	de ce pont.		
Souvent	je	m'accoude		au muret	de ce pont		

Effrayé	par	l'insecte,	je	rentre	précipitamment.
Effrayé	par	l'insecte	je rentre précipitamment.		
Effrayé par l'insecte,			je rentre		précipitamment
Effrayé par		l'insecte	je	rentre	précipitamment.

Ce	soir	nous	ne	nous	coucherons	pas	tard.
Ce	soir	nous ne		nous coucherons pas tard.			
Ce soir		nous ne nous coucherons				pas tard.	
Ce	soir	nous	ne nous coucherons			pas	tard

Nous	avons	pris	froid	en	jouant	au	tennis.
Nous	avons	pris	froid	en	jouant	au tennis.	
Nous avons pris froid				en jouant		au tennis.	
Nous	avons	pris	froid	en jouant au tennis			

C'est	un	charmant	spectacle	je	t'assure.
C'est	un	charmant spectacle		je t'assure.	
C'est un charmant spectacle,				je t'assure.	
C'est	un charmant		spectacle	je	t'assure.

Il	arrive	demain	d'Italie	par	la	route.
Il arrive		demain d'Italie par			la route.	
Il arrive demain			d'Italie	par la route.		
Il	arrive	demain	d'Italie par la route.			

Je	rends	souvent	visite	à	mon	oncle.
Je	rends	souvent visite à			mon oncle.	
Je rends		souvent visite		à mon oncle.		
Je	rends	souvent	visite	à mon oncle.		

Ma	partition	était	sous	ce	pupitre.
Ma	partition était		sous ce pupitre.		
Ma partition		était	sous ce pupitre.		
Ma partition		était	sous	ce pupitre.	

Ma	soirée	se	passera	sans	incident.
Ma	soirée	se	passera sans incident.		
Ma soirée		se passera		sans incident.	
Ma soirée		se passera		sans incident.	

TABLE A.1 – Phrases (et découpages) utilisées dans le test perceptif visant à évaluer le regroupement induit par la méthode proposée. Les phrases sont classées par ordre alphabétique et l'ordre des découpages est le suivant : mot-à-mot, aléatoire, expert, latence adaptative

Annexe B

Phrases du corpus de test pour l'évaluation subjective de la stratégie joker

---

Ci-après sont présentées les phrases utilisées pour l'évaluation subjective de la méthode de synthèse. Chacune des phrases a été synthétisée à l'aide des modèles de synthèse non-incrémentale, "Joker" et "Par Défaut". Ces phrases sont extraites du corpus issu du *tour du monde en 80 jours* de Jules Verne décrit en Section 3.5.1.1.

— C'était un homme qui avait dû voyager partout, en esprit tout au moins.
— Il ne perdait pas un regard au plafond, il ne se permettait aucun geste superflu.
— Puis il se fit servir à déjeuner dans sa cabine.
— Je ne suis pas sans avoir prévu l'éventualité de certains obstacles.
— On se mit à la besogne en faisant le moins de bruit possible.
— Il veillait à ce que rien ne manquât à la jeune femme.
— Il arrivât même que le jeune garçon allât plus loin un autre jour, mais c'était plus fort que lui.
— Il fallait bien en prendre son parti, et la terre ne fût signalée que le six, à cinq heure du matin.
— D'ailleurs nous arriverions pas à temps car il y a seize cent cinquante miles de Hongkong à Yokohama.
— Le lendemain, huit novembre, au lever du soleil, la goélette avait fait plus de cent miles.
— En France, on exhibe des farceurs étranger et à l'étranger, des farceurs français.
— Un acte d'extraction était maintenant nécessaire pour l'arrêter.

HMM TRAINING STRATEGY FOR INCREMENTAL SPEECH SYNTHESIS

Maël Pouget [1, 2], Thomas Hueber [1, 2], Gérard Bailly [1, 2], Timo Baumann[3]
[1] CNRS/GIPSA-Lab, Grenoble, France
[2] Univ. Grenoble Alpes/GIPSA-Lab, Grenoble, France
[3] Universität Hamburg, Informatics Department, Natural Language Systems, Hamburg, Germany
[1, 2] firstname.lastname@gipsa-lab.fr, [3] baumann@informatik.uni-hamburg.de

Abstract

Incremental speech synthesis aims at delivering the synthetic voice while the sentence is still being typed. One of the main challenges is the online estimation of the target prosody from a partial knowledge of the sentence's syntactic structure. In the context of HMM-based speech synthesis, this typically results in missing segmental and suprasegmental features, which describe the linguistic context of each phoneme. This study describes a voice training procedure which integrates explicitly a potential uncertainty on some contextual features. The proposed technique is compared to a baseline approach (previously published), which consists in substituting a missing contextual feature by a default value calculated on the training set. Both techniques were implemented in a HMM-based Text-To-Speech system for French, and compared using objective and perceptual measurements. Experimental results show that the proposed strategy outperforms the baseline technique for this language.

Index Terms: HMM-based speech synthesis, incremental, TTS, HTS, prosody

1. Introduction

Incremental Text-To-Speech (iTTS) systems aim at starting delivery of the synthetic voice before the full sentence context becomes available, e.g. while a user is still typing the text to vocalize. Contrary to a conventional TTS, the synthesis follows the text input, words after words (potentially with a delay of one word). This 'synthesis-while-typing' approach is illustrated in Figure 1. By reducing the latency between text input and speech output, iTTS should enhance the interactivity of communication. In particular, it should improve the user experience of people with communication disorders who use a TTS system in their daily life, as a substitute voice. Besides, iTTS could be chained with incremental speech recognition systems, in order to design highly responsive speech-to-speech conversion systems (for application in automatic translation, silent speech interface, real-time enhancement of pathological voice, etc.).

To our best knowledge, the concept of incremental speech synthesis was initially formulated in [1] in the context of dialogue systems. However, in the proposed proof-of-concept, the speech generation was delivered incrementally but was generated in a non-incremental way. In [2], Baumann & Schlangen proposed the first complete software architecture dedicated to incremental speech processing (including recognition, dialogue management and TTS modules). Another proof-of-concept based on the reactive HMM-based parameter generation system MAGE was also described in [3].

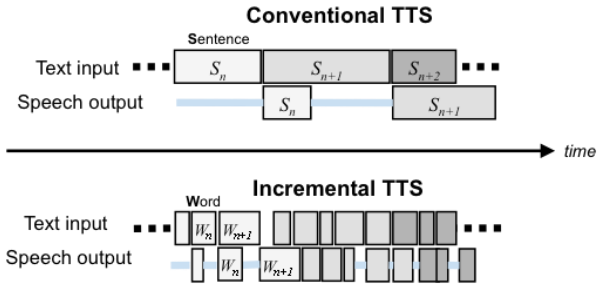


Figure 1: *Conventional versus incremental TTS*

One of the main remaining challenges of iTTS systems is the online estimation of the target prosody from an incomplete sentence (and therefore an uncertain - incrementally unveiled - syntactic structure). In conventional TTS, target prosody is typically calculated from long-range contextual features [4], [5], extracted from the text by morphological and syntactic analyzers. Considering a current segment as reference (typically a phoneme), some of these features refer to its left context (i.e. the 'past'); some of them refer to its right context (i.e. the 'future'). These features can, for instance, be the part-of-speech tag (POS) of the next word, or the number of remaining words before the end of the current sentence. Indeed, such features related to the right context are usually not available in incremental processing. Therefore, strategies should be developed to deal with these 'missing' features and predict acceptable prosody from an 'incomplete' sentence. This is the general scope of the present study.

In [6], [7], Baumann first evaluated the impact of potentially missing features on the quality of the estimated prosody, in the context of HMM-based speech synthesis, for English and German languages. Then, the author proposed a strategy for predicting a 'default' value for these missing features (this strategy is therefore referred here to as the 'Default' strategy). This strategy exploits the decision trees that are classically used in HMM-based speech synthesis in the state clustering procedure. The goal of this present study is twofold. First, we evaluate this strategy [6] (briefly recalled in Section 2) for French language, which has different prosodic characteristics than English and German (for instance, French can be considered to have no lexical stress[8]). Second, we propose another approach for dealing with missing contextual features, also in the context of HMM-based speech synthesis (Section 3). Contrary to [6], our approach does not aim at recovering the missing features at synthesis time. It rather consists in integrating a potential uncertainty on some features when building the synthetic voice (i.e. when training the HMM set). This strategy is here referred to as the 'Joker' strategy. The two

strategies were implemented in an HMM-based TTS for French language, and compared using objective measurements and a perceptual listening test (Section 4).

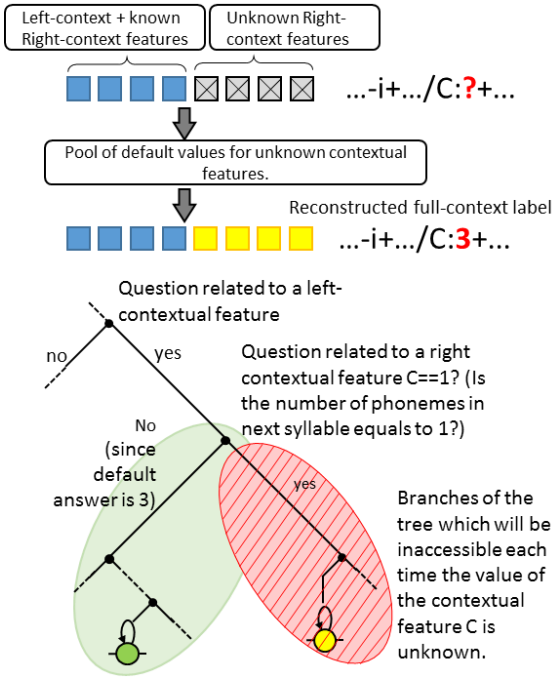


Figure 2. *Procedure for recovering full context labels from incomplete labels and clustering tree exploration when building unseen labels with the default strategy.*

2. Baseline strategy: 'Default'

In most implementations of HMM-based TTS (such as [9] or [10]), each speech unit is a phone in context. The context is described by a set of segmental features such as the identity of the current and adjacent phonemes, and suprasegmental features, such as the POS of the current and adjacent words, the position of the word in the current breath group, etc. Since it is very difficult to build a training dataset covering all possible contexts, clustering techniques are used to group some HMM states and share their model parameters (similarly to ASR systems). The most widely used technique is tree-based clustering [11]. Each node of the tree is associated with a context-related question, such as 'R-SYLL-NB-PHON==3' ("Are there 3 phonemes in the next syllable?"). The pertinence of the context-related questions and the structure of the tree are learnt automatically from the training dataset with respect to a specific criterion, such as the Minimum Description Length [12]. At synthesis time, the decision tree is extensively used when building the so-called 'unseen models' (i.e. corresponding to contexts with no acoustic observations in the training set).

In [6], Baumann proposed to exploit these decision trees to recover the missing contextual features at synthesis time. The procedure, which is illustrated in Figure 2, can be summarized as follows. First, a set of full-context HMMs (i.e. HMMs modeling each phoneme with information about its left and right contexts) is trained using a standard procedure (including tree-based clustering). Then, a 'default answer' is assigned to each contextual feature related to the right context (which might be unknown in incremental processing). This 'default answer' is calculated from the training set, by averaging the answers observed at each node of the decision tree associated with a question on the right context. For numerical features (e.g. the number of words before the end of the sentence), the default answer is the mean value calculated across the training dataset; for symbolic features (e.g. the POS tag of the next word), the default answer is the most common value observed in the training dataset.

This strategy gives encouraging results and works with pre-existing voices that were not trained with the incremental use-case in mind, but presents a major disadvantage. By imposing a default value to some contextual features, some branches of the clustering trees become totally unexplored when building the 'unseen models' (as shown by the red dashed zone in Figure 2). Therefore, only a limited number of HMM states are used at synthesis time. In other words, the fine-grained modeling of the training dataset based on a rich set of contextual features is not exploited here. In the next section, we present another strategy to deal with missing contextual features in the context of HMM-based incremental synthesis.

3. Proposed strategy: 'Joker'

In the proposed approach, the potential uncertainty on right-contextual features is handled during voice training rather than during the synthesis process, as in the 'Default' strategy. The proposed technique aims at considering a contextual feature that could potentially be missing as 'relevant' information that can be explicitly used when describing the linguistic context. Besides, when clustering the pool of HMM-states, we evaluate the need of tying model parameters among all contexts potentially sharing the same missing features. This training procedure results in a set of context-dependent HMMs that are likely to be slightly less accurate than full-context models (for instance, they are expected to deliver a neutral intonation for situations where the right context would trigger very different patterns). However, the 'Joker' strategy may lead to better perceptual results since there is no risk it uses an incorrect full-context model, as in the Default' strategy.

The 'Joker' has been implemented in the HTS framework as follows. First, the training corpus is labeled by introducing a so-called 'Joker' value (specified by the # character) to each contextual feature which cannot be determined when processing the text incrementally. This notably affects all the contextual features requiring information about the next word. As an example, let us consider the label associated with the phoneme in the last syllable of the current word, and the right-contextual feature 'number of phonemes in the next syllable (usually denoted by the symbol '*C*' in HTS). Since the value of this feature is unknown when processing the text incrementally, the 'Joker' tag is inserted in the label such as: "…-p+…/C:#..." (other contextual features are omitted for clarity). Then, a set of context-dependent HMMs is trained using a standard procedure (similarly to a non-incremental system). The Joker tag (#) is simply considered as a possible value for some contextual features.

A tree-based clustering procedure is then applied to deal with data sparsity. However, contrary to a non-incremental system, we introduce questions about the possible known/unknown characteristic of each contextual feature. In the HTS format, this can be written as: "*QS "R_nb_phone_in_next_syllabe_is_unknown" {*/C:#}"* where *C* stands for the number of phonemes in the next syllable. At the end of the clustering procedure, the parameters of some models/states sharing a common missing feature are expected to be tied together. The rest of the training procedure, as well as

the synthesis procedure are similar to a non-incremental system. As shown in Figure 3, one of the main advantages of the 'Joker' strategy compared to the 'Default' strategy, is a better utilization of the decision tree when building the unseen models. Even if a question related to a missing contextual feature in used at a specific node of the tree, the label can continue to 'go down' to all sub-branches of this node. Therefore, in this case, all HMM states are reachable. This should result in more variety in the estimated trajectories, compared to the 'Default' strategy, as shown by the experimental results described in the next sections.

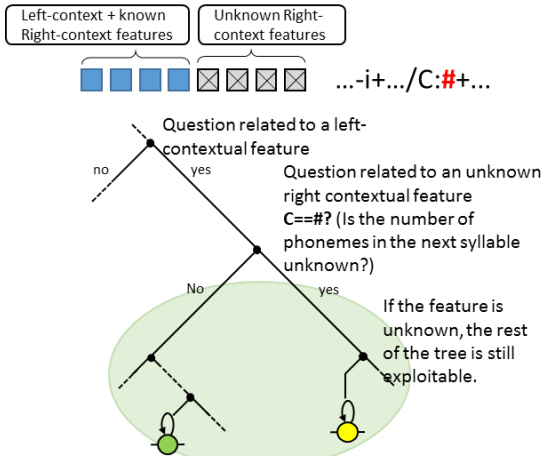


Figure 3. *Usage of the decision tree for building unseen models using the 'Joker' strategy.*

4. Experimental protocol and results

4.1. HMM-based TTS system for French language

The two strategies described in the previous sections were evaluated in the context of an HMM-based TTS system, developed in our group for French language. The specificities of this system are the following. The audio material used for training was extracted from an audiobook of the novel "Le tour du Monde en 80 jours" by Jules Verne (this corpus was also used in [13]). This corpus contains 3h17mn of speech data, after silence being removed. Phonetization as well as morphological and syntactic analyses of the text transcriptions were achieved using the linguistic front-end COMPOST [14]. The contextual features considered in this study are listed below (the features which are potentially missing in an incremental scenario are written in bold):

- Identity of the n-2, n-1, n (current), **n+1**, **n+2** phoneme
- Position of current phoneme in the current syllable (forward & backward)
- Number of phonemes in the previous/current/**next** syllable.
- Identity of the vowel of the current syllable
- Position of the current syllable in the word (forward & backward)
- Position of the current syllable in the sentence (forward & **backward**)
- POS-tag of previous/current/**next** word
- Number of syllables in the previous/current/**next** word
- Position of the current word in the sentence (forward & **backward**)
- **Sentence type** (assertion, wh-question, full question, etc.)

An initial segmentation of the audio recordings at phonetic level was obtained using a forced-alignment procedure and was then post-processed manually. In our system, the full-band spectral envelope is parameterized using a "Harmonic plus Noise Model" (HNM) [15] (and not mel-cepstral coefficients as in [16]), following the implementation detailed in [17] (p.82). Each acoustic observation (extracted each 5 ms) is a 93-dimensional vector composed of the fundamental frequency f_0, a (12^{th}-order) LSF-modeling of the harmonic component of the spectral envelope (defined for voiced frames only), and a (16^{th}-order) LSF-modeling of the residual spectrum, completed by first and second derivatives. A set of context-dependent HMMs (5 emitting states for the acoustic models) were trained on this corpus using the HTS toolkit [9] and a standard procedure. Global variance optimization was not used in this study.

Similarly to [6], we calculated the percentage of use of each right-contextual questions for clustering the training set, for spectrum-related streams, pitch and duration. As shown in Figure 4, we observed approximately the same pattern for French and German (see Figure 4 in [6]). As in [6], most of the questions recruited for clustering the spectrum-related parameters were related to the quinphone context. However, for pitch and duration, more questions related to current and next word were used for French than for German.

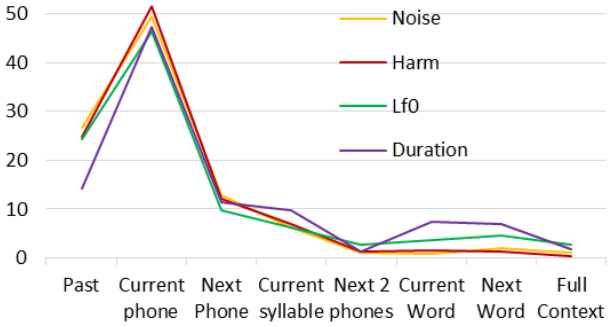


Figure 4. *Percentage of use of right-contextual questions used in the decision tree for clustering the training set, for spectrum (Harmonic+Noise), pitch and duration*

4.2. Objective evaluation

The two strategies considered in this study were first evaluated using as set of objective measurements. A subset of 165 test sentences was randomly selected from the corpus (and removed from the training set). These sentences were first synthesized using a non-incremental approach. The resulting acoustic feature vectors (i.e. spectrum, f_0 and duration) were considered as the 'best possible result'. In other words, we assume that incremental processing will systematically lead to lower performance than non-incremental processing (a similar assumption was made in [6]). The 165 test sentences were then synthesized using both 'Default', and 'Joker' strategies. The accuracy of the estimated spectrum was evaluated by calculating a mel-cepstral distortion [18] in dB defined such as:

$$MCD(\mathbf{y}_t^S, \mathbf{y}_t^{NI}) = \frac{1}{T}\frac{10}{\ln(10)}\sum_{t=1}^{t=T}\sqrt{2\sum_{d=0}^{D}(\mathbf{y}_t^S - \mathbf{y}_t^{NI})^2} \quad (1)$$

where $\mathbf{y}_t^S$ and $\mathbf{y}_t^{NI}$ are respectively vectors of *D+1* mel-ceptral coefficients estimated using the *S* incremental strategy and the baseline non-incremental (*NI*) approach and *T* the number of frames in the utterance. These coefficients were derived from the HNM-model of the spectrum (section 4.1) using the SPTK

toolkit [19]. A perceptual-based measure [20] of the difference (in cents) (also used in [21]) between incremental and non-incremental approaches in terms of f_0 was calculated for each utterance such as:

$$E_{f_0} = \frac{1}{T}\sum_{t=1}^{T} 1200 \log_2 \left| f_0^S(t) / f_0^{NI}(t) \right| \quad (2)$$

The timing distortion induced by non-incremental processing was evaluated by calculating for each test sentence the log duration ratio [22] such as:

$$E_{dur} = \frac{1}{P}\sum_{p=1}^{P} \log(\mathrm{d}_{\mathrm{NI}_p} / \mathrm{d}_{\mathrm{S}_p}) \quad (3)$$

where $\mathrm{d}_{\mathrm{NI}_p}$ and $\mathrm{d}_{\mathrm{S}_p}$ are the estimated phoneme duration obtained using non-incremental and incremental approaches, respectively, and P is the number of phonemes in the utterance. For each metric (*MCD*, E_{f0}, E_{dur}), the statistical significance of the difference between 'Default' and 'Joker' strategies was assessed using a paired t-test. Experimental results are presented in Table 1.

Table 1: Objective differences between 'Default' and 'Joker' for spectrum, f0 and phone duration, averaged across the test set (± standard deviation).

	Default	Joker	Joker vs. Default
MCD (dB)	0.78 ± 0.26	0.94 ± 0.15	***
E_{f0} (cents)	197.4 ± 88.7	178.2 ± 78.4	NS
E_{dur}	0.20 ± 0.06	0.17 ± 0.04	***

In terms of spectrum estimation, the distortions observed with both incremental strategies are relatively small (less than 1 dB). This result is compatible with the study of [23] showing that a look ahead of two phonemes (i.e quinphone modeling with no long-range contextual features) is enough to accurately estimate the target spectrum. The highest distortion was obtained with the 'Joker' strategy (which can therefore be considered as slightly less accurate than the default strategy). However, the difference between the two strategies, even statistically significant, is tiny (0.16 dB). An opposite effect was observed for pitch and segment duration (which are more closely related to prosody). Also, and despite statistical significance, the differences between the two strategies remain too small to conclude to a perceptual difference (i.e. with less than 20 cents for *f0*). Therefore, a listening test was conducted to study in more detail potential perceptual differences between the two strategies.

4.3. Perceptual evaluation

The perceptual evaluation was conducted with a ranking listening test, similar to [24] and [25]. For each trial, the subject was asked to sort 3 sound samples, according to its "naturalness". These sound samples correspond to the same sentence synthesized respectively with the non-incremental approach, the 'Default' approach, and the proposed 'Joker' approach. Two stimuli used in this test are submitted as supplementary material (*exampleX_S.wav* with *X*={*1,2*}, *S*={*joker*, *default*, *nonIncremental*}). For each test sentence, the user interface used for this test was composed of a ranking X/Y area, in which each listener was asked to 'drag and drop' the 3 audio samples to rank. Each sample was represented by a 'push button' allowing to listen to it, as many times as required. The X-axis of the ranking area was a continuous scale (ranging from 0 to 5). A set of 5 labels ("very bad, bad, middle, good, very good") was nevertheless added to help the subject in the ranking process (the scale can thus be considered as semi-continuous). The position on the Y-axis was not taken into account (as told to the subject). The test was conducted in a quiet room with the same headphones, with 18 native speakers of French, with no particular expertise in speech synthesis. They were asked to evaluate a set of 12 sentences (resulting in 36 stimuli in total). These sentences were randomly extracted from the test set (the shortest selected sentence was 14 syllables long, the longest was 27 syllables long). For each trial and each participant, the presentation order of the stimuli was randomized. Both parametric (ANOVA) and non-parametric (Kruskal-Wallis) tests were conducted to assess the statistical significance of the results. These tests considered the X-position on the ranking area as the continuous variable to explain, the 3-level explicative variable *Strategy* (with the possible values 'non-incremental', 'default', 'joker'), and a random *Listener* effect on the intercept. Since the main effect of the factor *Strategy* was significant ($p<0.005$), post-hoc analyses were conducted to test the contrast between 'Default' and 'Joker' strategies. Results are presented in Figure 5.

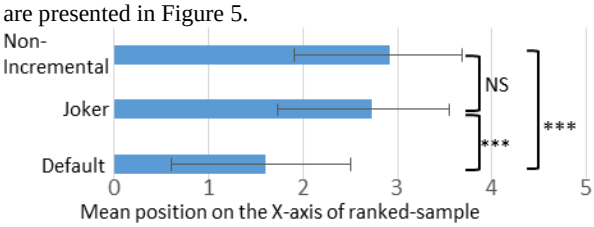


Figure 3. Results of the perceptual listening test: mean position on the X-axis of ranked-sample, averaged across the listeners, for both non-incremental and incremental ('Default' and 'Joker') strategies.

As expected, the best-ranked samples were those obtained with the non-incremental approach (i.e. with a complete set of contextual features). The proposed 'Joker' strategy outperforms significantly the baseline ('default') strategy (2.8 vs. 1.5, $p<0.005$). This supports the benefit of considering explicitly the uncertainties about right context when building the synthetic voice. Interestingly, no statistically significant difference was observed between the non-incremental and the 'Joker' strategy. Amongst other possible causes, this could be explained by a 'ceiling effect', due to the intrinsic quality of the baseline HMM-based synthesis (with a mean score of 3).

5. Conclusion and perspectives

This study describes a strategy for dealing with missing contextual features, for incremental HMM-based speech synthesis. This strategy consists in integrating a potential uncertainty on some contextual features when training the HMM set. This approach was compared to the baseline technique proposed in [6]. Both strategies were implemented in an HMM-based TTS system for French. A perceptual test shows that the proposed strategy outperforms the baseline technique for that language. Future work will focus on the evaluation of the proposed strategy for other languages, such as English and German (which were considered in [6]). In order to build a complete incremental TTS system, we will also combine the proposed technique with and 'incremental text parsing' front-end. Such module could be inspired by some approaches developed for incremental text processing and syntactic parsing [26].

6. References

[1] J. Edlund, "Incremental speech synthesis," in *Proceedings of Swedish Language Technology Conference*, Stockholm, Sweden, 2008, pp. 53–54.

[2] T. Baumann and D. Schlangen, "The INPROTK 2012 release," in *Proceedings of NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, Stroudsburg, PA, USA, 2012, pp. 29–32.

[3] M. Astrinaki, N. d' Alessandro, and T. Dutoit, "MAGE: A Platform for Performative Speech Synthesis New Approach in Exploring Applications Beyond Text-To-Speech," in *Proceedings of The Listening Talker Workshop*, Edinburgh, Scotland, 2012, p. 53.

[4] D. Büring, "Syntax, information structure and prosody," in *The Cambridge Handbook of Generative Syntax*, Marcel den Dikken, 2013, pp. 860–895.

[5] C.-Y. Tsai, C.-K. Kuo, Y.-R. Wang, S.-H. Chen, I.-B. Liao, and C.-Y. Chiang, "Hierarchical prosody modeling of English speech and its application to TTS," in *Proceedings of Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA), 2014 17th Oriental Chapter of the International Committee for the*, Pukhet, Thaïlande, 2014, pp. 1–6.

[6] T. Baumann, "Decision tree usage for incremental parametric speech synthesis," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 3819–3823.

[7] T. Baumann, "Partial Representations Improve the Prosody of Incremental Speech Synthesis," in *Proceedings of Interspeech*, Singapore, 2014.

[8] M. Rossi, *Le Français, langue sans accent?*, Studia Phonetica Montréal., vol. 15. 1980.

[9] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of ICASSP*, Istanbul, Turkey, 2000, vol. 3, pp. 1315–1318.

[10] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proceedings of Eurospeech*, Budapest, Hungary, 1999, pp. 2347–2350.

[11] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based State Tying for High Accuracy Acoustic Modelling," in *Proceedings of the Workshop on Human Language Technology*, Stroudsburg, PA, USA, 1994, pp. 307–312.

[12] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn.*, vol. 21, no. 2, pp. 79–86, 2000.

[13] G. Bailly and C. Gouvernayre, "Pauses and respiratory markers of the structure of book reading," in *Proceedings of Interspeech*, Pittsburgh, PA, USA, 2006.

[14] M. Alissali and G. Bailly, "COMPOST: a client-server model for applications using text-to-speech systems.," in *Proceedings of European Conference on Speech Communication and Technology*, Berlin, Germany, 1993, pp. 2095–2098.

[15] Y. Stylianou, "Harmonic plus Noise Models for Speech combined with Statistical Methods, for Speech and Speaker Modification," Ecole Nationale superieure des télécommunications, Paris, France, 1996.

[16] "The HTS toolkit." [Online]. Available: http://hts.sp.nitech.ac.jp/.

[17] T. Hueber, "Reconstitution de la parole par imagerie ultrasonore et vidéo de l'appareil vocal : vers une communication parlée silencieuse," Université Pierre et Marie Curie, Paris, France, 2009.

[18] R. F. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of Communications, Computers and Signal Processing, IEEE Pacific Rim Conference on*, Victoria, British Columbia, Canada, 1993, vol. 1, pp. 125–128 vol.1.

[19] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proceedings of ICASSP*, San Francisco, CA, USA, 1992, vol. 1, pp. 137–140 vol.1.

[20] I. Peretz and K. L. Hyde, "What is specific to music processing ? Insights from congenital amusia," *Trends Cogn. Sci.*, vol. 7, no. 8, pp. 362–367, 2003.

[21] M. Astrinaki, N. d' Alessandro, B. Picart, T. Drugman, and T. Dutoit, "Reactive and continuous control of HMM-based speech synthesis," in *Proceedings of Spoken Language Technology Workshop (SLT), IEEE*, Miami, FL, USA, 2012, pp. 252–257.

[22] N. W. Campbell, "Segment durations in a syllable frame," *J. Phon.*, vol. 19, no. 1, pp. 37–47, 1991.

[23] S. Le Maguer, "Evaluation experimentale d'un systeme statistique de de la parole, HTS, pour la langue francaise.," Université de Rennes 1, Rennes, France, 2013.

[24] G. Bailly and I. Gorisch, "Generating German intonation with a trainable prosodic model," in *Proceedings of Interspeech*, Pittsburgh, PA, USA, 2006, pp. 2366–2369.

[25] H. R. Pfitzinger, "Local Speech Rate As A Combination Of Syllable And Phone Rate," in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998, pp. 1087–1090.

[26] N. Beuck, A. Köhn, and W. Menzel, "Predictive incremental parsing and its evaluation," in *Computational Dependency Theory*, vol. 258, Kim Gerdes, Eva Hajičová, Leo Wanner, 2013, p. 186.

Adaptive Latency for Part-of-Speech Tagging in Incremental Text-to-Speech Synthesis

Maël Pouget[1,2], *Olha Nahorna*[1,2], *Thomas Hueber*[1,2], *Gérard Bailly*[1,2]

[1]CNRS/GIPSA-Lab, Grenoble, France
[2]Univ. Grenoble Alpes/GIPSA-Lab, Grenoble, France
[1,2]firstname.lastname@gipsa-lab.fr

Abstract

Incremental text-to-speech systems aim at synthesizing a text 'on-the-fly', while the user is typing a sentence. In this context, this article addresses the problem of the part-of-speech tagging (POS, i.e. lexical category) which is a critical step for accurate grapheme-to-phoneme conversion and prosody estimation. Here, the main challenge is to estimate the POS of a given word without knowing its 'right context' (i.e. the following words which are not available yet). To address this issue, we propose a method based on a set of decision trees estimating online whether a given POS tag is likely to be modified when more right-contextual information becomes available. In such a case, the synthesis is delayed until POS stability is guaranteed. This results in delivering the synthetic voice in word chunks of variable length. Objective evaluation on French shows that the proposed method is able to estimate POS tags with more than a 92% accuracy (compared to a non-incremental system) while minimizing the synthesis latency (between 1 and 4 words). Perceptual evaluation (ranking test) is then carried in the context of HMM-based speech synthesis. Experimental results show that the word grouping resulting from the proposed method is rated more acceptable than word-by-word incremental synthesis.

Index Terms: Incremental speech synthesis, natural language processing, classification, TTS, part-of-speech

1. Introduction

Text-to-speech (TTS) systems are now able to produce very high-quality synthetic voice. They can be used as a substitute voice by people with severe communication disorders (such as patients with Parkinson's disease or ALS). However, TTS-based communication lacks interactivity since the synthesis is generally triggered on a per-sentence basis. Therefore, the listener (i.e. the communication partner) has to wait for a complete sentence to be typed down. This increases drastically the communication latency and often results in some frustration for both the listener and the system user. Incremental TTS (iTTS) [1, 2, 3] aims at improving this interactivity issue by delivering the synthetic voice 'on-the-fly' (*i.e.* while the user is typing the target sentence) with almost the same quality as a conventional (*i.e.* non incremental) TTS.

The main challenge in iTTS is to perform the two main steps of a conventional TTS, that are text analysis (often referred to as natural language processing, NLP) and waveform generation, when considering only a limited lookahead. In other word, the iTTS paradigm assumes that the synthesis of a given word can rely only on its 'left-context' (*i.e.* the words before it) and that almost no 'right-context' (*i.e.* the words after it) is available. In our previous study [4], we focused on the waveform generation step in the context of HMM-based speech synthesis. We proposed a method for building HMM voices using models trained with limited and adaptive lookahead. In this paper, we focus on the text analysis step, and in particular on Part-Of-Speech (POS) tagging. This step consists in assigning a lexical category to each word (e.g. noun, verb, etc.), based on both morphological analysis and syntactic constraints, *i.e.* its relationship with left- and right-adjacent words in the sentence. POS-tagging is critical for grapheme-to-phoneme conversion but also for prosody estimation since the syntactic structure of the sentence is actually derived from the POS tags.

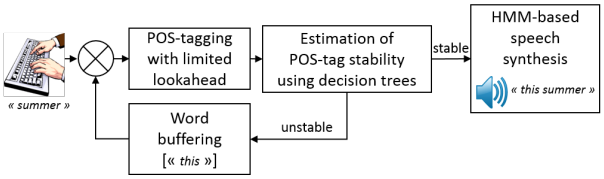


Figure 1: *Overview of the proposed iTTS architecture with adaptive latency for robust online POS-tagging*

In [5], Beuck et al. propose four strategies for performing POS-tagging incrementally, in the context of NLP. These strategies as well as their use in the context of iTTS can be briefly summarized as follows:

- estimating POS tags using left-context only. In iTTS, this results in a zero delay for delivering the synthetic voice but some POS tags may be inaccurate.
- considering a fixed size lookahead (typically 2 or 3 words) for disambiguating POS tags. In iTTS, this results in a constant latency but likely more accurate POS tags.
- recalculating the POS of a given word already tagged when more right-context becomes available (a system allowing such behavior is referred by the authors as a non-monotonic system). In iTTS, the synthesis has to be postponed until the POS tag can no longer be modified. As discussed later, this is the core idea of the iTTS architecture proposed in this article.
- considering multiple hypothesis for each new available word. In iTTS, this will require to propagate such ambiguities to the signal processing module. This approach seems interesting but is not considered in the present study.

In line with the third strategy, we propose a method for estimating POS tags accurately in the context of iTTS while

minimizing the lookahead (and thus maximizing the reactivity of the synthesizer). The proposed method (described in Section 2) is based on a set of decision trees estimating online whether a given POS tag is likely to be modified when more right-contextual information becomes available. Each decision tree models the stability of a POS tag for a given left-context and a given lookahead. In the present study, we consider an adaptive lookahead between 0 and 2 words. The synthesis of a word is triggered as soon as the stability of its related POS-tag is guaranteed. This results in delivering the synthetic voice in word chunks of variable length (i.e. adaptive latency). A general overview of the proposed architecture for a so-called "adaptive-latency iTTS" is presented in Figure 1. The proposed method is evaluated both objectively and perceptively, in the context of our HMM-based iTTS system for French [4] (Section 3).

2. Proposed method

2.1. POS-tagging in incremental TTS

Many approaches have been proposed in the literature to address automatic POS-tagging in conventional (i.e. non-incremental) TTS (see [6], ch. 10 and [7] for reviews). Modern taggers are almost all based on the two following steps: (1) the extraction of one or several hypothesis for each word considered separately from its context and (2) a global optimization which aims at alleviating ambiguities by making use of the large-span context. POS-tagger such as TnT [8] or Festival [9] use second order Markov models with states representing the tags and outputs (i.e. observations) representing the words (and thus state transition probabilities modeling pairs of tags). The POS-tagger used in this study for French language, called COMPOST [10], is based on the same approach. Following the formulation used in [8], the most likely tag sequence $[\hat{c}_1, \ldots, \hat{c}_T]$ associated with the word sequence $[w_1, \ldots, w_T]$ of length T is defined such as:

$$\underset{[c_1,\ldots,c_T]}{\arg\max}\{[\prod_{t=1}^{T} P(c_t|c_{t-1}, c_{t-2})P(w_t|c_t)]P(c_{T+1}|c_T)\} \quad (1)$$

where c_{-1}, c_0, and c_{T+1} are beginning/end sentence markers, $P(w_t|c_t)$ is related to tag estimation without taking into account any contextual information and $P(c_t|c_{t-1}, c_{t-2})$ refers to a 3-gram model providing prior information on the current tag c_t given the tags of the two previous words (c_{t-1} and c_{t-2}). These probabilities can be derived from relative frequencies estimated on large text corpora. In the framework of Markov modeling, Equation (1) is typically solved using the Viterbi algorithm.

Such formulation assumes that the final tag c_{T+1} is known without any ambiguity (as well as c_{-1} and c_0). In conventional TTS, it often corresponds to a "End-of-sentence" marker such as a period. However, such assumption can not be made when processing the input text incrementally. Thus, the POS-tagging technique needs to be adapted. Here, we propose to solve (1) for the word sequence $[w_1, \ldots, w_t]$ each time a new word w_t is made available (e.g. when the user presses the space bar), using the forward-backward rather than the Viterbi algorithm. The associated tag c_t is defined as the one that maximizes the forward probabilities $\alpha_t(j,k) = P(c_1, \ldots, c_{t-1} = j, c_t = k|w_1, \ldots, w_t)$ over all the possible N tags. This forward probability can be calculated using the well-known recursive expression:

$$\alpha_t(j,k) = \sum_{i=1}^{N} \alpha_{t-1}(i,j)P(c_t = k|c_{t-1} = j, c_{t-2} = i) \quad (2)$$

assuming that word w_{t-2} was given the tag i and a transition between states j and k at times $t-1$ and t. Each previous word w_k of $[w_1, \ldots, w_{t-1}]$ is then tagged by calculating the posterior probabilities:

$$P(c_k = k|w_1, \ldots, w_t) = \sum_{j=1}^{N} \alpha_k(j,k)\beta_k(j,k) \quad (3)$$

with $\beta_k(j,k)$ the backward probability given by:

$$\beta_t(j,k) = \sum_{i=1}^{N} \beta_{t+1}(i,j)P(c_t = k|c_{t+1} = j, c_{t+2} = i) \quad (4)$$

2.2. Evaluation of POS tag stability using decision trees

The POS-tagging procedure presented in the previous section is sub-optimal since an uncertainty remains on the final tag c_t. Indeed, its online estimation relies only on the left-context and therefore may sometimes be incorrect. Moreover, if c_t is incorrect, the backward propagation may influence in a bad way the tags further left (i.e. $[c_1, \ldots, c_{t-1}]$). To alleviate this potential negative effect, we propose a method for estimating the stability of a POS tag, in a given (syntactic) context, that it how it is likely to be modified when more right-context becomes available.

The proposed method is based on a set of 3 binary decision trees. Each decision tree models the stability of a POS tag for a given lookahead (i.e. right-context) of 0,1,or 2 words. Input features are composed of a sequence of 3 consecutive tags $[c_{t-2}, c_{t-1}, c_t]$ calculated incrementally, together with their associated probabilities $[P(c_{t-2} = i|w_1, \ldots, w_t), P(c_{t-1} = j|w_1, \ldots, w_t), P(c_t = k|w_1, \ldots, w_t)]$ given by Equation (3). The output feature is a binary value indicating if the POS tag calculated incrementally matches the one estimated from the complete left and right context. In other word, for each tree, the set of yes/no questions partitions the training set regarding the following rules: where L is the considered lookahead and T the number of words in each training sentence. Note that c_{-1} and c_0 are set as an explicit Beginning-of-Sentence class with a probability equal to 1. As an example, let us consider the French sentence "*Cet été, les enfants vont à la mer*" ("This summer, the children will go to the sea"). Training input observations are built by successively sending the following chunks to the POS-tagger: "*Cet*", "*Cet été,*", "*Cet été, les*", "*Cet été, les enfants*", "*Cet été, les enfants vont*", etc. and by storing the successive POS tags for each word, with their respective probabilities.

2.3. Adaptive latency iTTS

As already mentioned, a POS-tagging error can have important consequences on the grapheme-to-phoneme conversion as well as on the prosody. With this consideration in mind, we propose a new iTTS architecture in which the synthesis of a given word w_t is delayed until the stability of its associated POS-tag (determined using the procedure describe in Section 2.1) is guaranteed. This stability is assessed using the decision trees presented in Section 2.2. The proposed algorithm for triggering the synthesis is presented in Algorithm 1. This procedure results in delivering the synthetic voice in word chunks of variable length, introducing a variable latency but maximizing the POS-tagging accuracy. The maximum latency which can be obtained using this procedure is 3 words. This happens when a given POS tag is still classified as "unstable" even when considering a 2-words lookahead.

Data: $[w_{t-2}, w_{t-1}, w_t]$, $[c_{t-2}, c_{t-1}, c_t]$
waiting list : Typed words, not synthesized yet.
if w_{t-3} *is in waiting list* **then**
  $Synthesize(w_{t-3})$
if w_{t-2} *is in waiting list* **then**
  if $IsStable(c_{t-2})$ *(2-word lookahead)* **then**
    $Synthesize(w_{t-2})$
  else
    Put w_{t-2}, w_{t-1}, w_t in waiting list **return**;
if w_{t-1} *is in waiting list* **then**
  if $IsStable(c_{t-1})$ *(1-word lookahead)* **then**
    $Synthesize(w_{t-1})$
  else
    Put w_{t-1}, w_t in waiting list **return**;
if w_t *is in waiting list* **then**
  if $IsStable(c_t)$ *(0-word lookahead)* **then**
    $Synthesize(w_t)$
  else
    Put w_t in waiting list **return**;

Algorithm 1: Proposed algorithm for scheduling the incremental synthesis of chunks of words based on the stability of the POS-tagging.

3. Experiments

3.1. Objective evaluation

The proposed method was evaluated in the context of our incremental HMM-based speech synthesis system [4], which is based on the NLP front-end COMPOST [10] and the HTS toolkit [11]. The corpus used for training the decision trees was extracted from the two French books "Notre-Dame de Paris", by Victor Hugo and "Le tour du monde en 80 jours", by Jules Verne. This corpus consists in 20154 sentences (290801 words). The corpus was divided into a training set (2/3 of the corpus : 13436 sentences, around 193000 words) and a testing set (1/3 of the corpus : 6718 sentences, around 98000 words). The training of the decision trees was done using Matlab (*classregtree* package).

First, we evaluated the performance for each of the 3 decision trees considered *independently*. That is, their ability to evaluate whether a POS tag is likely to be modified when considering more right-context (i.e. a lookahead of 0, 1, or 2 words). The performance was measured by calculating the *accuracy* (Acc), defined as

$$Acc = (TP + TN)/(TP + TN + FP + FN)$$

where TP, TN, FP, and FN are respectively true positives, true negatives, false positives and false negatives. Results are presented in Figure 2.

First, let us discuss the performance in terms of POS tag correctness as a function of the lookahead (that is the raw performance of the NLP front-end COMPOST considered in this study). With no lookahead, around 40% of the POS tag are badly estimated (i.e. $(TN+FP)/(TP+FN+TN+FP)$). As expected, the performance increases with the lookahead, with 9% of error when considering 1 word, and less than 2% when considering 2 words. These results show that (1) POS-tags can be accurately estimated online when considering at least a lookahead of two words, (2) a new strategy was in fact needed to achieve lower latency. Let us now discuss the ability of the decision trees to evaluate the stability of a POS tag.

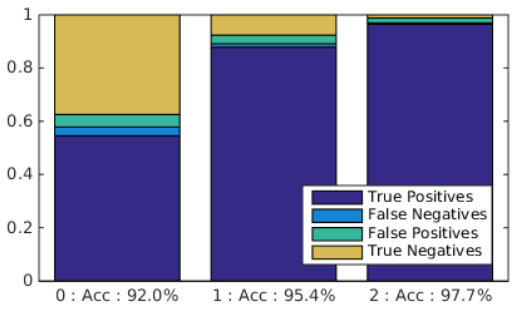


Figure 2: *Objective evaluation of the decision trees (considered independently) estimating the stability of a POS tag as a function of the lookahead (for left to right: 0, 1, and 2 words).*

With no lookahead, the stability of the POS tag was correctly assessed in 92% of the cases (i.e. $(TP + TN)/(TP + TN + FN + FP)$). Among these decisions, in 37% of the cases, it was rightly decided to postpone the synthesis since the stability of the POS was not guaranteed (TN). On the contrary, in 55% of the cases, the POS tag was considered to be stable so that the synthesis could be triggered confidently (TP). In 8% of the cases, the stability of the POS tag was wrongly assessed, resulting either in a synthesis triggered too soon and with an erroneous POS tag (FP) or with an unnecessary latency (FN). As expected, the number of such errors (i.e. $FP + FN$) decreases when the lookahead increases, with $\sim 4\%$ for a 1-word lookahead and $\sim 2\%$ for a 2-word lookahead.

Then, we evaluated the performance of the complete system, that is when the 3 decision trees are used jointly as shown in Algorithm 1 (in other words, the decision of the "no lookahead tree" conditions the decision of the "1-word lookahead tree", etc.). Figure 3 displays the distribution of the test data as a function of the delay needed to guarantee the POS tag stability. For each considered lookahead (0, 1 and 2, resulting in a maximum latency of 3 words), we also represent the remaining errors (FP, in yellow), that is the amount of words for which the synthesis has been wrongly triggered instead of being delayed. In 60% of the cases, the synthesis is triggered immediately (no lookahead) with 92% of the POS tag correctly estimated. In more than 30% of the cases, a lookahead of 1-word is needed to estimate the POS-tag with 95.4% accuracy. Finally, in 5% of the cases, the synthesis is delayed by at least 2-words (with more than 97.7% accuracy). When considering the combined accuracy of all decisions performed with a maximum latency of 3 words, the proposed adaptive latency approach performs a robust online POS-tagging with $\sim 90\%$ correlation with respect to the non incremental tagging.

3.2. Perceptual evaluation

The proposed iTTS system with adaptive latency delivers the synthetic voice in groups of words (between 1 and 4 words). This may result in a singular word grouping (i.e prosodic phrasing). To assess the quality of this grouping, we conducted a perceptual evaluation based on a ranking test. A set of 14 sentences extracted from the Combescure corpus [12] was synthesized using our HMM-based iTTS system for French [4] and 4 different strategies of word grouping (resulting in a total of 56 stimuli to rank):

- "WG1: One word per group" which corresponds to a

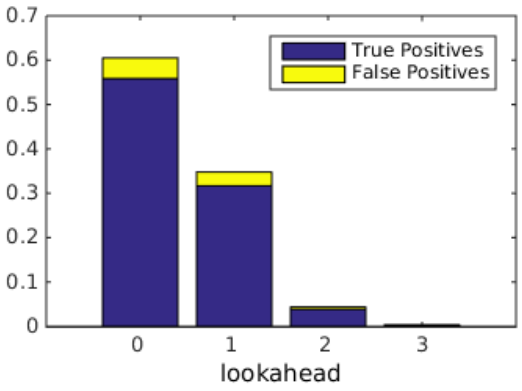


Figure 3: Distribution of the test words as a function of the lookahead needed to guarantee the stability of the associated POS tags.

word-by-word synthesis using no lookahead (e.g "This - summer, - the - children - etc.")

- "WG2: Random word grouping" obtained by replacing the output of each decision tree by a random binary value. This strategy is used as a reference condition (e.g "This - summer, the - children - etc.").
- "WG3: Expert-based word grouping" where 3 human experts were asked to delimit manually the most natural boundaries of each prosodic phrase, based on the semantic (e.g "This summer, - the children - will go - to the sea").
- "WG4: Adaptive latency iTTS" which is the word grouping resulting from the proposed method (e.g "This summer, - the children will - go to - the sea").

The duration of the silence between each word group is constrained so that the 4 versions of each sentence have all the same length (with minimum silence duration between each word chunk set arbitrarily to 300 ms). The listening test was done online by 20 native speakers of French, with no particular expertise in speech processing. The participants were asked to do the test in a quiet environment, with headphones. The presentation order of the stimuli was randomized for each participant. For each sentence, the participant were asked to score the different stimuli on a Mean-Opinion-Score (MOS) scale ranging from 1 to 5 (a set of 5 labels "very bad, bad, middle, good, very good" was nevertheless displayed in order to help the subject in the ranking process). The participant was allowed to play each stimulus several times. The statistical significance of the ranking score was assessed using Beta regression, considering the position of the stimulus on the scale as the variable to explain, the word grouping strategy as the explanatory variable (4-level factor), and both the *subject ID* and *sentence ID* as random effects (an Anova test was not suitable since the variable to explain was bounded).

As expected, the most natural word grouping is the one proposed by human experts (WG3), which can indeed rely on high-level semantic knowledge. Interestingly, the "one word per group" strategy (i.e. the strategy that leads to the most reactive system) was considered less acceptable than the random grouping (which was the reference condition). This result shows the importance of prosodic phrasing in incremental text-to-speech, where a tradeoff between reactivity and naturalness have to be found. Finally, and more importantly, the proposed adaptive-latency iTTS was ranked second. It was assessed significantly

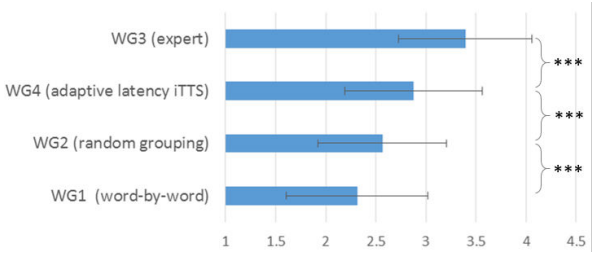


Figure 4: *Results of the perceptual listening test. Mean position on the X-axis of ranked samples with standard deviations, averaged across the listeners, for each word grouping strategy ("one word per group" (WG1), "random" (WG2), "expert" (WG3) and "adaptive-latency iTTS" (WG4, proposed method)(*** denotes statistical significance)*

.

better than the random grouping (and word-by-word synthesis), but also significantly lower than the expert-based strategy. This demonstrates the interest of the proposed approach while letting some room for improvements. To illustrate a possible limitation of the proposed method, let us focus on one stimulus which was ranked as "bad" by most listeners: the sentence "Il arrive en retard en ce moment" ("he arrives late these days"). For this stimulus, the expert-based word grouping (WG3) was *"Il arrive - en retard - en ce moment"* whereas the word grouping resulting from the analysis of POS tag stability gave *"Il arrive - en retard en - ce moment"*. This result in a non-natural prosodic phrasing, notably due to the third chunk "en retard en". It corresponds to the POS sequence "Preposition Noun Preposition" which is not a common prosodic unit in French. Therefore, the proposed word grouping strategy based on the sole POS-tag stability is an interesting but perfectible approach.

4. Conclusions and Perspectives

This article introduced a method for robust POS-tagging in the context of incremental Text-to-speech synthesis. The core idea is to assess 'on-the-fly' whether a POS tag in a given left-context is likely to be modified when more right-context becomes available, and if yes, to postpone the synthesis. This results in a new iTTS architecture where the synthetic voice is delivered in word chunks of variable length. Objective evaluation showed that almost 90% accuracy of true positives can be obtained with a adaptive lookahead between 0 and 3 words, for French.

Although demonstrating the pertinence of this morphosyntactic parsing for effective incremental speech synthesis, the perceptual evaluation of the resulting prosodic phrasing led to contrasting results. Future work will focus on improving this prosodic phrasing. Among other perspectives, we will notably combine the proposed approach with the predictive incremental parsing technique, recently proposed in [13]. Finally, as an incremental TTS synthesizer is primarily designed for casual conversation, we will also evaluate the performance of the proposed adaptive latency POS-tagger on other kind of text data, such as text-messages or tweets.

5. Acknowledgments

This work was funded by the project *SpeakRightNow* (AGIR program, Université Joseph Fourier, http://www.gipsa-lab.fr/projet/SpeakRightNow/). The authors would like to thank Sylvain Gerber for his help in the statistical analyses.

6. References

[1] J. Edlund, "Incremental speech synthesis," in *Proceedings of Swedish Language Technology Conference*, Stockholm, Sweden, 2008, pp. 53–54.

[2] D. Schlangen and G. Skantze, "A general, abstract model of incremental dialogue processing," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009, pp. 710–718.

[3] T. Baumann and D. Schlangen, "The INPROTK 2012 release," in *Proceedings of the 12th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA, 2012, pp. 29–32.

[4] M. Pouget, T. Hueber, G. Bailly, and T. Baumann, "HMM training strategy for incremental speech synthesis," in *Proceedings of Interspeech*, Dresden, Germany, Sep. 2015, pp. 1201–1205.

[5] N. Beuck, A. Köhn, and W. Menzel, "Decision Strategies in Incremental PoS Tagging," in *Proceedings of NODALIDA 2011*, Riga, Latvia, 2011, pp. 26–33.

[6] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[7] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, Stroudsburg, PA, USA, 2000, pp. 63–70.

[8] T. Brants, "TnT: a statistical part-of-speech tagger," in *Proceedings of the sixth conference on Applied natural language processing*. Seattle, WA, USA: Association for Computational Linguistics, 2000, pp. 224–231.

[9] P. Taylor, A. W. Black, and R. Caley, "The Architecture of the Festival Speech Synthesis System," in *Proceedings of the 3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998, pp. 147–151.

[10] G. Bailly and M. Alissali, "Compost : un serveur de synthèse de parole multilingue," *Traitement du Signal*, vol. 9, no. 4, pp. 359–366, 1992.

[11] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech Synthesis Based on Hidden Markov Models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, May 2013.

[12] P. Combescure, "listes de dix phrases phonétiquement équilibrées," *Revue d'acoustique*, vol. 56, 1981.

[13] N. Beuck, A. Köhn, and W. Menzel, "Predictive incremental parsing and its evaluation," in *Computational Dependency Theory*. Kim Gerdes, Eva Hajičová, Leo Wanner, 2013, vol. 258, p. 186.