UNIVERSITÉ JOSEPH FOURIER - GRENOBLE I
SCIENCES & GÉOGRAPHIE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER - GRENOBLE I

Discipline : Informatique Système et Communication

présentée et soutenue publiquement

par

Dominique Vaufreydaz

le 7 janvier 2002

Titre :

Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue

Directeur de thèse : Jean Caelen

JURY

M. Christian Boitet	*Président*
M. Renato De Mori	*Rapporteur*
M. Kamel Smaïli	*Rapporteur*
M. Laurent Besacier	*Examinateur*
M. Célestin Sedogbo	*Examinateur*

Remerciements

Tout d'abord je voudrais remercier *Jean Caelen*, mon directeur de thèse, pour m'avoir accueilli au sein de l'équipe GEOD du laboratoire CLIPS pendant ces années d'études et pour m'avoir encadré pendant cette thèse. Je remercie aussi *Laurent Besacier* pour sa participation active à cette thèse en tant que co-encadrant.

Je remercie tous les membres de mon jury, c'est-à-dire *Renato De Mori* et *Kamel Smaïli* pour avoir accepté d'être rapporteurs de ma thèse, *Célestin Sedogbo* et *Laurent Besacier* pour en avoir été examinateurs et *Christian Boitet* pour sa présidence.

Toute ma gratitude à toutes les personnes ayant relu, corrigé et commenté mon manuscrit et ayant ainsi participé à son amélioration.

Je remercie mes parents pour m'avoir toujours poussé dans mes études. Je remercie aussi ma grand-mère, mon frère *Franck*, ma belle-soeur *Christine*, *Harmonie* et aussi la petite *Tiphaine* pour leur soutien au cours de ces années.

Je profite de cette page pour remercier l'ensemble des membres du laboratoire pour leur accueil, et plus particulièrement les membres des équipes GEOD et MULTICOM que j'ai côtoyés quotidiennement avec un réel plaisir pendant ces années. Je remercie également le personnel administratif du laboratoire pour son efficacité et sa bonne humeur.

Je tiens à saluer aussi toutes les personnes de l'IUP Métiers du livre, à savoir *Lysiane*, *Rose*, *Cécile*, *Évelyne*, *Éric* et *Jean-Marc*, où j'ai le plaisir d'avoir été ATER en informatique.

Je remercie *Jean-François Serignat* pour m'avoir délégué certaines responsabilités pour lesquelles j'espère avoir été à la hauteur. Je tiens à remercier *Mohammad Akbar* pour avoir su distiller son savoir et pour la somme colossale de connaissances qu'il m'a transmises. Il en est de même pour *Bernard Cassagne*. Je n'oublie pas non plus la contribution de *Christian Boitet* par les très nombreuses discussions que nous avons eues au cours de ces années et pour m'avoir souvent éclairé de ses connaissances encyclopédiques.

Je remercie la « Dream Team » c'est-à-dire *Mohammad*, « *pépé* » et *Hervé* avec qui j'ai passé de si bonnes soirées pour préparer les démonstrations des projets, ainsi que les autres membres du laboratoire participant activement à CSTAR et Nespole!, *Laurent* et *Jean-Philippe* (dont la prose n'a d'égale que la très grande sagesse philosophique).

Une mention très spéciale à tous mes amis du laboratoire c'est-à-dire *Anne*, *Jean-*

François, *Laurent*, *Richard* (et *Daphné*), *Carole*, *Solange*, *Mathias*, *Hatem*, *Yannick*, *Brigitte*, *Éric* avec qui j'ai énormément apprécié de travailler mais aussi de ne pas travailler ! Je remercie *Hélène, Marie-Laure, Bruno dit « papy », Marika, Mireille, Vincent, Yann et d'autres qui m'ont soutenu et accompagné au cours de ces années.*

J'exprime toute ma reconnaissance à *Bernard* et *Nicolas* sans lesquels je n'aurais très certainement pas pu terminer ce doctorat. Il en est de même pour les familles *Ginet* et *Bonnardon* dont le soutien a très fortement contribué à la réussite de cette thèse. Je souhaite à tout le monde de pouvoir compter sur le soutien sans faille de personnes de cette qualité.

Côté musical, je remercie aussi mes amis de l'orchestre inter-école et Monsieur *Bernard Sémino,* son chef. J'ai eu la chance de ne pas apprendre que la musique de ce dernier. Je remercie aussi *Nicolas* et *Yves* avec qui j'ai pris un réel plaisir à animer des soirées, des bals et des mariages pendant très longtemps. Je n'oublie pas mes amis de Titch Ka Ra, c'est-à-dire *Séverine*, *Lionel* et *Franck*, pour leur bonne humeur et le plaisir que je prends à jouer et composer des morceaux avec eux. Cette dédicace s'adresse aussi aux anciens membres du groupe comme *Annabelle*, *Claire*, *Laurent* et *Michel*. Je les félicite d'avoir su supporter mon humour pendant nos très nombreuses répétitions.

Enfin, pour leur soutien musical tout au long de cette thèse et de sa rédaction, je remercie en vrac *Eric Clapton*, *Zazie*, *Eddy Mitchell*, *Claude Nougaro*, *Sting*, *Queen*, *The Corrs*, *Popa Chubby*, et de trop nombreux autres...

Dédicace

Cette thèse est dédiée à ma maman, Jany, décédée lors de ma première année de thèse, qui m'a toujours poussé et motivé dans mes études. Sans elle, je n'aurais certainement pas fait d'études longues. Cette thèse représente donc l'aboutissement du soutien et des encouragements qu'elle m'a prodigués tout au long de ma scolarité. Qu'elle en soit remerciée par cette trop modeste dédicace.

Préambule

Cette thèse a été rédigée en HTML à l'aide d'un éditeur de texte simple. L'utilisation de *Cascading Style Sheets* (feuilles de styles) a permis l'obtention d'un rendu similaire aux documents réalisés avec des éditeurs avancés. Ce choix, volontaire, avait pour but de fournir une version correcte, au niveau présentation, de notre manuscrit de thèse pour l'exposé sur la Toile. Grâce à notre expérience du langage HTML, nous avons écrit plusieurs outils de numérotation automatique des titres, de construction automatique de tables des matières, etc. Cependant, pour des raisons techniques indépendantes de notre volonté, nous n'avons pu empêcher certains problèmes de mise en page comme par exemple les veuves et les orphelines. Nous tenons à nous en excuser auprès des lecteurs de ce manuscrit.

Table des matières

Liste des équations

Liste des exemples

Liste des figures

Liste des tableaux

Introduction

Introduction

- bonsoir Dave
- ça va HAL ?
- tout est en ordre de marche, et vous ça va ?
- pas trop mal
- je vois que vous avez travaillé...
- quelques croquis...
- je peux les voir ?
- bien sûr
- c'est très bien rendu Dave... je crois que vous avez fait beaucoup de progrès... un peu plus près je vous prie
- bien sûr [...]

Cet extrait de dialogue est tiré de « *2001, l'odyssée de l'espace* », l'adaptation cinématographique du premier tome de la célèbre tétralogie de Arthur C. Clarke réalisée par Stanley Kubrick. Voilà l'avenir tel que l'auteur du livre le dépeignait dans les années soixante. Un ordinateur capable de comprendre le langage oral, d'interagir directement avec les humains en utilisant la parole comme modalité d'entrée et de sortie. Cela allait même bien plus loin puisque celui-ci était finalement animé de capacités supérieures qui lui permettaient d'apprendre la lecture labiale afin d'espionner ses passagers et d'assouvir ses pulsions paranoïaques. C'est ainsi qu'était HAL, un ordinateur personnifié comme une tierce personne dans un futur qui est aujourd'hui notre présent. Clarke n'était bien entendu pas seul à avoir cette vision de l'avenir. Nombre d'auteurs de science-fiction avaient prédit que l'ordinateur prendrait une place prépondérante dans la société et que son intégration, dans la vie quotidienne, serait complète. Alors qu'en est-il aujourd'hui à l'aube du 21ème siècle et du 3ème millénaire ? Bien évidemment, les progrès techniques ont suivi et même dépassé, dans beaucoup de domaines, et notamment dans le médical, les prévisions les plus optimistes. En ce qui concerne la reconnaissance de la parole, il faut reconnaître que, malgré les progrès énormes réalisés ces dernières années, les chercheurs ne peuvent pas prétendre encore combler les espérances des auteurs de science-fiction. En cela, nous suivons les conclusions de [Stork 99] sur les différences entre HAL et les systèmes actuels.

Pourtant, de nos jours, la reconnaissance automatique de la parole est de plus en plus utilisée, grâce notamment à l'arrivée de produits commerciaux grand public de bonne qualité et financièrement accessibles. Mais l'engouement pour cette technologie pourrait pourtant masquer les problèmes qu'il reste à résoudre dans ce domaine de recherche. Comme le

soulignait déjà en 1986 Bristow [Bristow 86], « Speech recognition is about computers learning how to communicate with humans, rather than vice versa ». il est évident que si l'effort d'adaptation vient de l'utilisateur plutôt que du système : le problème se pose en des termes contraires à la notion d'utilisabilité. Or, la plupart des systèmes actuels nécessitent non seulement une phase d'apprentissage acoustique, mais aussi une utilisation contrainte du système pour l'obtention de résultats de bonne qualité. Cette contrainte porte sur l'utilisation d'une certaine forme de langage pour que le système de reconnaissance parvienne à le traiter. En 2000, [Gauvain 00b], dans son état de l'art de la reconnaissance de la parole et indiquait qu'il y avait encore de très nombreux progrès à réaliser pour que les technologies vocales puissent être utilisables dans des conditions réelles. Sa conclusion est que, même si de nombreuses avancées ont été réalisées ces dernières années dans tous les domaines d'étude liés à la reconnaissance de la parole, il est difficile de savoir quelle partie de nos systèmes doit être encore améliorée.

Nous avons choisi de nous intéresser dans notre thèse à la partie modélisation du langage et ceci, plus particulièrement, pour les systèmes de reconnaissance de la parole spontanée à grand vocabulaire. Pour ce faire, nous avons choisi d'utiliser des modèles statistiques. Dans cette optique, il est nécessaire de disposer de corpus d'apprentissage de taille suffisante pour l'estimation des probabilités de nos modèles. De plus, ceux-ci doivent être représentatifs du langage qui sera utilisé dans l'application, dans notre cas, le langage oral. L'un des problèmes majeurs est alors la constitution de ces corpus. Les approches de type *magicien d'Oz* ne permettant pas l'obtention de données en quantité suffisante, nous avons tenté d'utiliser des documents en provenance d'Internet comme corpus de départ. Nos hypothèses étaient que l'on y trouverait des formes propres au langage oral et cela en quantité suffisante pour l'apprentissage de nos modèles. L'objectif de notre thèse est de fournir les outils et les méthodes de filtrage qui permettent, à partir de ces données très bruitées, d'obtenir automatiquement des corpus et de pouvoir calculer un modèle de langage statistique propre à modéliser des énoncés en langage naturel.

Le manuscrit est composé de trois parties principales. La première partie introduit le contexte de notre étude et un état de l'art de la reconnaissance de la parole. Son premier chapitre permet de comprendre quel est l'intérêt de la modélisation du langage au sein d'un système de reconnaissance. Nous y expliquons quelles étaient les motivations qui nous ont poussé à utiliser des modèles de langage statistiques. Nous abordons aussi le problème du manque de corpus d'apprentissage du fait de la difficulté de leur collecte. Nous introduisons enfin *CSTAR* et *Nespole!*, deux projets de traduction de parole spontanée qui nous ont servi de cadre d'expérimentation en conditions réelles de nos modèles de langage.

Le second chapitre présente un état de l'art de la reconnaissance de la parole. Il débute par la présentation de l'approche entièrement probabiliste en reconnaissance automatique de la parole. Il se poursuit par la description du signal de parole et des paramètres que l'on en extrait. Puis, il introduit la modélisation acoustique à base de modèles de Markov cachés qui est, à l'heure actuelle, la méthode la plus employée en reconnaissance de la parole continue. Ce chapitre se termine par la présentation des algorithmes permettant l'obtention, sous une forme textuelle, du résultat final du système de reconnaissance de la parole.

Le chapitre III décrit la modélisation du langage par modèles probabilistes. Nous y présentons les techniques les plus répandues de nos jours comme, par exemple, les modèles n-grammes ou n-classes et quelques-unes de leurs variantes. Nous y énonçons le problème du manque de données d'apprentissage et présentons les techniques mises en œuvre pour remédier à ce problème et notamment l'approche de Katz.

La seconde partie de ce manuscrit concerne nos travaux sur la modélisation automatique du langage en utilisant des documents en provenance d'Internet. Dans le chapitre IV, nous décrivons nos critères de sélection parmi les données présentes sur Internet et les données que nous avons finalement collectées. Nous réalisons ensuite une étude quantitative mais aussi qualitative de ces documents, pour montrer leur adéquation à notre tâche de modélisation statistique du langage oral. Nous présentons aussi une étude comparative, portant sur le premier corpus extrait de l'Internet et sur le plus récent. Elle montre que l'intérêt du contenu de ces documents, pour la modélisation du langage, va croissant.

Le chapitre V introduit notre méthodologie de construction automatique de modèle de langage. On commence par l'extraction du texte, depuis nos corpus extrait d'Internet, en tenant compte de leur nature et de leur structure. On peut ensuite étendre le vocabulaire par l'ajout des mots les plus présents sur la Toile. Ensuite, un filtre construit, avec ce texte et le vocabulaire spécifié, ce que nous nommons des "blocs minimaux", qui nous permettent ensuite de constituer un corpus d'apprentissage. Avec des outils modifiés pour prendre en compte la forme de ce corpus, il est enfin possible de calculer un modèle de langage n-gramme.

Dans le chapitre VI, nous présentons les résultats que nous obtenons avec les modèles de langage construits avec notre méthode. Nous commençons par la description des métriques que nous employons pour leur évaluation en terme de perplexité et de couverture linguistique, puis dans le cadre de la reconnaissance de la parole. Nous étudions les résultats, en taux de reconnaissance, sur différentes tâches allant de 2500 mots à plus de 20000 mots. Nous comparons aussi nos résultats à ceux obtenus par d'autres équipes de recherche sur les signaux d'une évaluation des systèmes de dictée vocale proposée par l'*AUPELF-UREF*. Nous arrivons à des résultats très probants en comparaison avec les meilleurs actuels. Nous avons en particulier fourni la preuve qu'un gros corpus écrit, brut, venant du Web donne de meilleurs résultats que des corpus de parole transcrits spécifiques mais de petite taille.

Cependant, avec de très gros vocabulaires, les résultats sont encore perfectibles. Dans ce cas, l'utilisation d'un module de détection de thème peut conduire à un accroissement du taux de reconnaissance. Le chapitre VII décrit la première version d'un outil de détection de thème basé sur les *newsgroups* français et son intégration dans les algorithmes de reconnaissance. Notre objectif est d'utiliser une arborescence de thèmes pour déterminer, au cours d'un seul énoncé, le thème courant. Le principe est alors de modifier les probabilités d'un modèle de langage généraliste pour refléter le thème trouvé. Nous présentons les premiers résultats encourageants que nous obtenons avec cette technique.

Le chapitre VIII présente l'adaptation de notre méthode de collecte et de filtrage des documents en provenance d'Internet permettant la constitution d'un ensemble d'énoncés pour l'enregistrement d'un corpus sonore, donc l'amélioration des performances de notre module acoustique. Nous montrons que 75% des énoncés ainsi obtenus ne nécessitent pas de correction *a posteriori*. De plus, en terme de fréquence des phonèmes, le corpus sonore final est représentatif du français tel qu'il avait été étudié dans la littérature.

Nous terminons ce manuscrit par la présentation de nos perspectives concernant l'utilisation d'Internet dans le cadre de la modélisation multilingue statistique du langage en vue de la compréhension et de la traduction de parole.

Partie I : Contexte d'étude et état de l'art

Chapitre I : Contexte d'étude

Chapitre I : Contexte d'étude

Présentation du chapitre

La reconnaissance de la parole vise à faciliter la communication non seulement entre personnes et système, mais aussi entre humains. Il est donc important de traiter complètement les aspects essentiels, et notamment cognitifs, mis en jeu lors de l'émission d'un message et de la réception de celui-ci par l'interlocuteur visé. Nous soulignerons donc les différences entre le mode de communication humain/humain et le mode humain/machine. Nous introduirons la notion de modèle de langage utilisé par un ordinateur pour interpréter un message venant d'un être humain.

Nos travaux ont un objectif essentiel : améliorer la qualité des systèmes de reconnaissance de la parole. Cela n'a de sens que dans un cadre expérimental nous permettant de valider nos recherches, car il est impossible de construire, sans support pratique, un système de reconnaissance de la parole. Nous présenterons donc les buts des projets de recherche dans lesquels nous nous sommes intégré pour mener à bien nos travaux. Pour finir, nous décrirons précisément nos objectifs, en fixant un cadre précis de recherche à notre thèse.

I. Communication

I.1. Entre humains

Le langage oral est le mode de communication privilégié entre les êtres humains. Cela peut s'observer si l'on compare le nombre de langues parlées au nombre de langues écrites. Ainsi recense-t-on un peu plus de 5000 langues orales pour quelques centaines de langues écrites. Bien que certaines cultures aient d'ailleurs fondé la transmission de leur savoir sur une tradition orale, l'écrit possède une fonction de mémorisation avant tout. Ce fait est énoncé dans un dicton d'origine latine « *les paroles s'envolent, les écrits restent* ».

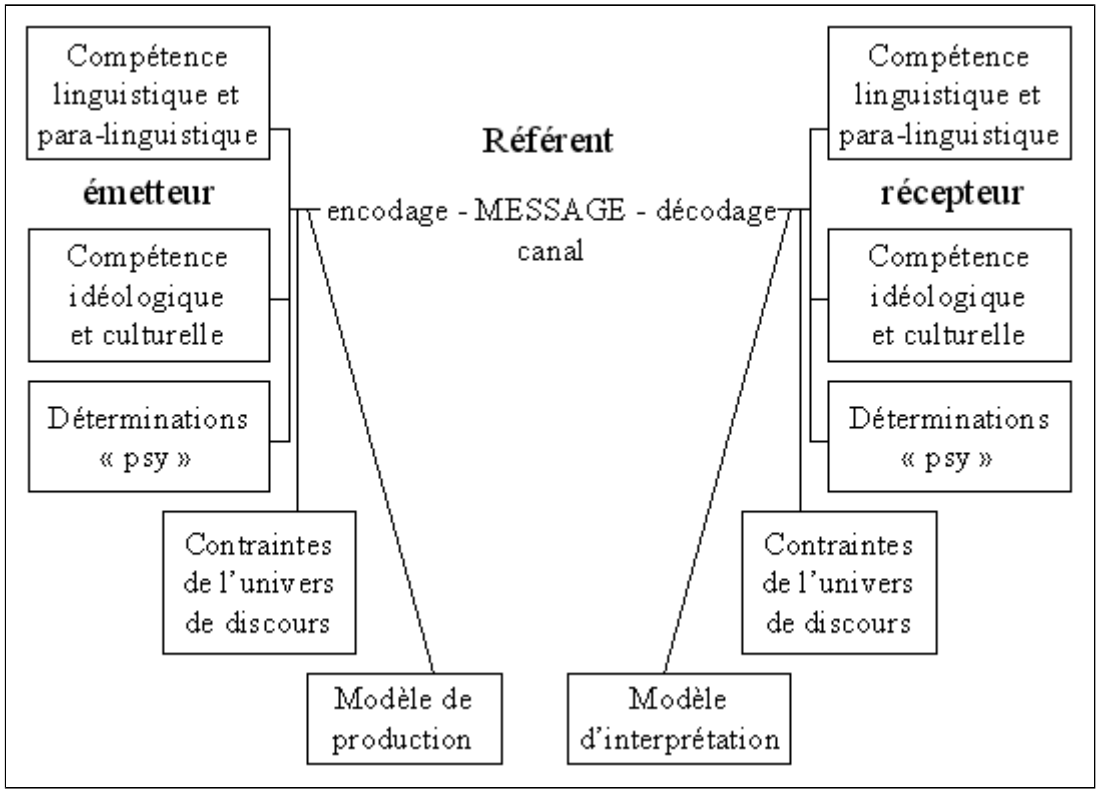


Figure I.1 : Modèle de communication entre humains (source [Kerbrat 80])

Le modèle de Kerbrat-Orecchioni ci-dessus, l'un des modèles disponibles, présente la transmission d'un message d'un être humain à un autre. Il est dérivé des travaux de Jakobson de la fin des années 50 [Jakobson 60]. La figure I.1 présente l'émetteur, le canal de transmission et le récepteur. Le canal fonctionne, comme le proposait Jakobson, en encodant l'information sous forme de message. Il y a ensuite transmission du message puis décodage par le récepteur. La première remarque très générale que l'on peut faire sur ce schéma concerne les mécanismes et les connaissances entrant en jeu lors de l'émission et la réception d'un message. Elles sont partagées pour tout ce qui concerne les connaissances linguistiques, paralinguistiques (gestes, attitudes, etc.), culturelles, idéologiques, et les contraintes de l'univers du discours. Les déterminations « psy » sont aussi utilisées par l'émetteur et le récepteur. Elles regroupent des fonctions déictiques dépendant de facteurs psychologiques et psychanalytiques comme le souligne [Kerbrat 80] page 18. Par exemple, si l'émetteur se trouve dans un état de colère, cela peut influer non seulement sur son message, mais aussi sur le décodage et l'interprétation du message par le récepteur. Selon que ce dernier est au courant ou non de l'état émotionnel de son interlocuteur, il peut alors décoder de façon correcte, ou non, un tel message. Tous ces éléments étant symétriques, il semble qu'il faille fournir autant d'efforts cognitifs pour produire que pour comprendre le message. Le modèle de production permet à l'émetteur, en fonction des connaissances de la langue de l'émetteur, de formuler au mieux son message en choisissant, par exemple, le meilleur énoncé pour exprimer une idée. De l'autre côté et dans le même temps, le récepteur doit sélectionner, pour les éléments du message porteurs potentiels de plusieurs sens, la signification adaptée au message grâce à son modèle d'interprétation.

D'autre part, il est clair que toutes les entités entrant en jeu dans la communication humaine ne sont pas des boîtes hermétiques. Les compétences culturelles et idéologiques ainsi que l'état psychologique influent sur les fonctions de production et d'interprétation. Les performances de ce système en font l'outil privilégié de la communication humaine.

I.2. Cas de la communication homme/machine ou homme/homme médiatisée

Comme nous venons de le voir, la richesse et la diversité des connaissances mises en œuvre dans la communication entre êtres humains font la robustesse de ce système. Dans le cas de la communication homme/machine ou homme/homme médiatisée, le schéma précédent est modifié du coté du récepteur qui est l'ordinateur. Il est difficilement envisageable d'insérer des connaissances culturelles et surtout idéologiques au sein d'un système de compréhension de la parole. En ce qui concerne les connaissances « psy », même si certaines études tentent de munir l'ordinateur de la capacité de construire un modèle de l'utilisateur [Ponton 96] pour mieux interagir avec lui, cela ne peut être réellement considéré comme de l'intégration de connaissances et d'états psychologiques dans l'analyse du message en provenance de l'utilisateur.

La première brique d'un système de compréhension de la parole, celle qui nous intéresse dans nos travaux, est le système de reconnaissance. Son but est de pouvoir déterminer, à partir des sons émis par l'utilisateur, quelle est la séquence de mots associée. L'ordinateur ne possède pour cela des contraintes sur les contraintes de l'univers du discours, les compétences linguistiques, et d'autres canaux dans le cadre d'une application multimodale. L'univers du discours correspond à la situation de communication (orale, écrite, monologue, dialogue, émise dans un cadre professionnel ou personnel, etc.) augmentée de règles stylistiques. Si l'on excepte les informations paralinguistiques, l'appellation générique pour ces connaissances dans un système de reconnaissance de la parole est *modèle de langage*. Cet outil permet au programme de choisir parmi un ensemble d'hypothèses jugées pertinentes, acoustiquement parlant, à un moment *t*. Pourtant, sur la base d'un modèle de langage, sans connaissance sémantique, il est quelquefois impossible même pour un humain de choisir une solution. Si l'on prend comme exemple, les phrases "*autant pour moi*" (1) et "*au temps pour moi*" (2), complètement identiques acoustiquement, il faut connaître le sens ou le déduire du contexte pour faire un choix. Dans le cas de (1), il y a notion de quantité alors que (2) représente une excuse [Grevisse 82]. C'est en partie pour cela qu'il est actuellement impossible de produire un système de reconnaissance donnant 100% de bons résultats dans toutes les conditions.

Les recherches menées dans le cadre de la modélisation du langage ne sont pas récentes [Rosenfeld 00]. Pourtant, même si de nombreuses améliorations ont été apportées au cours de la dernière décennie dans le domaine de la modélisation du langage, augmentant considérablement les performances des systèmes, il reste encore des progrès à réaliser. Les principaux problèmes qu'il reste à résoudre aujourd'hui, concernent la robustesse et la portabilité des modèles de langage, surtout pour des applications à très grand vocabulaire. C'est dans cette thématique que se placent nos travaux de recherche.

II. Communication homme/homme médiatisée multilingue

II.1. CSTAR phase II et III

Le laboratoire CLIPS est membre du consortium CSTAR [Web 01] (*Consortium for Speech Translation Advanced Research*) depuis 1995. Ce projet international regroupe des centres de recherche de nombreux pays : Carnegie Mellon University pour les USA, l'université de Karlsruhe pour l'Allemagne, les laboratoire ETRI en Corée, ATR au Japon et IRST en Italie. Le but des recherches menées au sein du consortium est la traduction automatique de parole spontanée avec tous les couples de langues possibles entre les différents partenaires. Dans sa phase II, la tâche visée était la réservation touristique et, plus précisément, la réservation de chambres d'hôtel, de billets d'avion ou de train, l'obtention de renseignements sur les horaires de ces transports et les principales activités touristiques possibles. Le scénario type est celui d'un client qui communique dans sa langue maternelle, par le biais du système de traduction, avec un agent de voyage qui ne parle pas la même langue que lui. Nous sommes donc dans le cas d'une communication homme/homme médiatisée.

Actuellement, la phase III du projet consiste en une version étendue de la tâche de réservation touristique : tous types d'information et de réservation touristique. La principale nouveauté concerne la traduction d'énoncés qui ne représentent plus simplement des actes de dialogues, comme par le passé, mais aussi des descriptions d'objets par exemple, ce qui est une tâche plus ardue pour la traduction. Une autre innovation est l'utilisation de téléphones portables de type GSM pour l'accès au service de traduction de parole. Cela introduit une difficulté supplémentaire au niveau du système de reconnaissance de la parole, qui doit pouvoir prendre en compte les dégradations dues à ce mode de transmission de la parole.

Pour simplifier la tâche, une méthode de traduction fonctionnant autour d'un langage pivot, l'IF pour *Interchange Format* [Levin et al. 98], un langage basé à l'origine sur l'expression d'actes dialogiques, réduit l'effort à fournir par chacun des partenaires, chacun ne s'occupant que de sa langue. Le schéma suivant montre le fonctionnement du système français et son intégration dans le système complet de traduction.

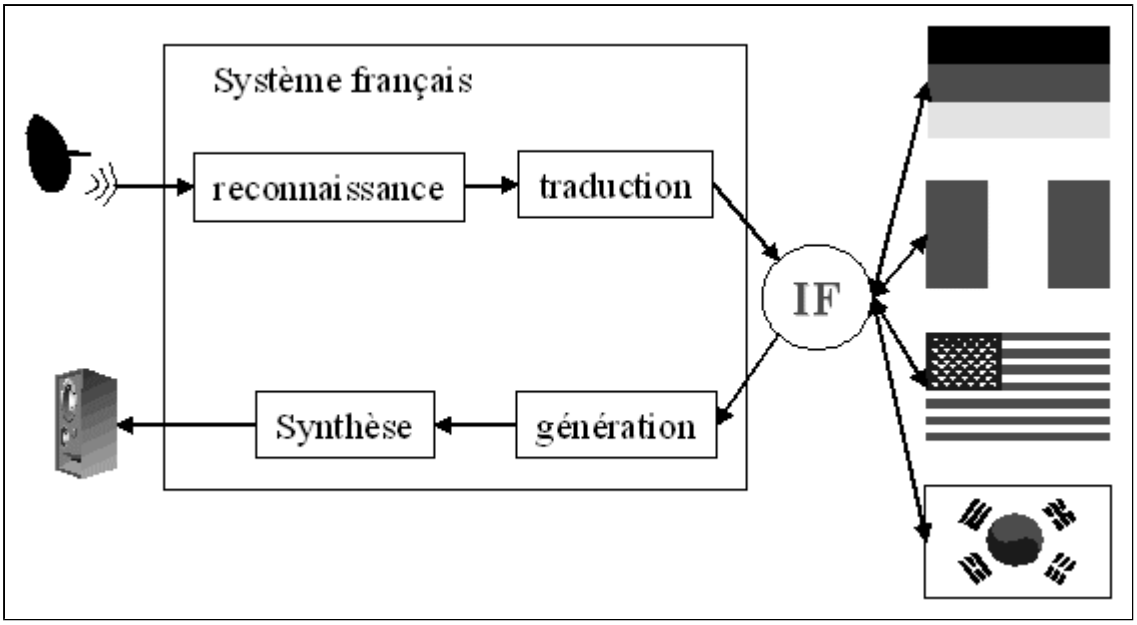


Figure I.2 : méthode de traduction par langage pivot du projet CSTAR

Comme nous le voyons sur ce synoptique, la traduction de parole consiste en une phase de reconnaissance d'un énoncé en langage naturel. Celui-ci doit ensuite être analysé et retranscrit en IF. Le message est ensuite envoyé vers les autres systèmes de traduction. Lors de la réception par le système français d'un message sous forme IF, celui-ci est transcrit en français avant d'être synthétisé. Chaque système, dans chaque langue, utilise le même procédé. Ainsi, avec une spécification commune de l'IF, il est possible de faire de la traduction de parole entre toutes les langues du consortium.

II.2. Nespole!

Le projet Nespole! [Web 02] (*Negotiating Through SPOken Language in E-commerce*) est un projet européen cofinancé par la communauté européenne et la NSF aux États-Unis pour permettre la collaboration de CMU avec ses partenaires européens de CSTAR. Dans un premier temps, le cadre de la traduction reste très proche du projet précédent : la réservation touristique. Cette base de travail doit permettre de résoudre les problèmes techniques avant d'aborder un autre domaine qui est de type aide téléphonique (*HelpDesk*). Nespole! intègre des outils de visioconférence pour la communication entre les deux utilisateurs avec, de plus, une interface multimodale permettant l'échange de données graphiques, comme par exemple des cartes, et la désignation d'objets sur celles-ci. Aethra [Web 10], une entreprise italienne, spécialisée dans les logiciels et protocoles de visioconférence, est partenaire du projet.

La traduction au sein du projet Nespole! utilise les mêmes mécanismes de langage pivot que CSTAR. Le premier objectif du projet est d'utiliser les données audio fournies par les systèmes de visioconférence, donc dégradées car codées, transmises sur Internet puis décodées, pour la reconnaissance de la parole. Le second vise à tester et mettre en œuvre la portabilité et l'extensibilité (*scalability* en anglais) de la méthode de traduction, et par là même l'extensibilité du langage IF à d'autres domaines que celui de la réservation touristique. Le dernier but est d'intégrer les données multimodales au sein de la traduction, par exemple

pour aider à la désambiguïsation, et de mesurer leur impact sur le dialogue entre le client et l'agent.

II.3. Intérêts de ces projets

Ces deux projets nous ont fourni un cadre d'expérimentation pour l'utilisation de nos modèles de langage. Nos travaux de recherche ont, en effet, été intégrés dès février 1999 dans le système de reconnaissance de la parole du CLIPS pour la préparation de la démonstration du projet CSTAR-II [Presse 99]. Par la suite, le projet Nespole! a été aussi une base d'essais enrichissante pour nos expériences. L'avantage principal était la possibilité de juger, dans des conditions réelles d'utilisation, de la pertinence de nos modèles dans un cadre dialogique. La mise au point de démonstrateurs et les nombreuses phases de test nous ont aussi permis d'analyser les problèmes et d'améliorer les méthodes et les outils que nous avions développés.

III. Définition du cadre de notre étude

III.1. Corpus

Pour cerner les énoncés que nous devons modéliser, nous avons conduit diverses actions dans le but d'obtenir des corpus de référence. Dans un premier temps, nous avons demandé à des chercheurs du laboratoire d'écrire des dialogues entre un client et un agent dans le cadre de la réservation touristique. Puis, nous avons utilisé des techniques de type *magicien d'Oz*, pour obtenir d'autres données. La totalité de ces données représente ce que nous appellerons dans le manuscrit le corpus « *réservations touristiques* ». Il comprend 12763 tours de paroles, soit 97253 mots. Dans un second temps, nous avons procédé à l'enregistrement de dialogues, dans des conditions réelles d'utilisation du système Nespole!, c'est-à-dire avec des outils de visioconférence. 5 scénarios différents, entre un client à Grenoble et un agent à Trento en Italie, ont été définis pour offrir une couverture maximum du domaine de la réservation touristique : informations sur des tours pré-organisés, organisation complète du séjour, renseignements sur les activités sportives, les parcs naturels, les festivals folkloriques et musicaux. Les consignes étaient simplement de suivre un scénario et de faire tout ce qui était nécessaire pour obtenir la réalisation de celui-ci. Cet enregistrement a été fait dans plusieurs langues pour finir sur un total de 191 réalisations dont 31 en français [Burger et al. 01]. Une transcription manuelle de ces dialogues a été effectuée et vérifiée. Ce corpus comprend 4104 tours de parole soit 42598 mots. À partir de maintenant, nous ferons référence à ce corpus sous le nom de « *corpus Nespole!* ».

Des exemples d'énoncés recueillis sont donnés dans l'exemple I.1. La transcription des tours de parole de l'agent est postfixée par 'A:'. Celle du client par 'C:'.

C: l'agence APT

A: oui bonjour c'est l'APT du... du (1) Trentino bonjour

C: oui bonjour je voudrais faire un voyage dans le Trentin

A: oui

C: et je voudrais savoir ce que vous proposez comme... comme (1) organisation toute prête avec les voyages les réservations d'hôtels

A: ah oui bien sûr nous avons des offres des forfaits pour l'hiver et pour l'été en quelle saison désirez-vous arriver

C: nous allons arriver au mois d'août donc en été

A: en été oui quelles exigences avez-vous vous avez(2) des des préférences pour certaines localités du Trentino

C: non non du tout c'est un voyage donc on sera deux adultes plus deux enfants

A: d'accord

Exemple I.1 : extrait d'un scénario du corpus Nespole!

Cet extrait est le début d'un dialogue conduisant à la réservation d'un séjour complet préorganisé par une agence de voyages. Cet exemple n'est ici que pour illustrer la difficulté de modélisation d'énoncés et n'est pas exhaustif quant aux phénomènes présents dans les dialogues oraux. À la lecture de ce dialogue, nous pouvons noter qu'il y a des hésitations provoquant des répétitions (notées 1). De plus, il est clair que le nombre de phrases est variable dans un même tour de parole (noté 2). Comme nous travaillons dans le cadre de systèmes de dialogue et non pas de dictée vocale, nous ne trouvons bien sûr aucune ponctuation qui pourrait servir à délimiter certaines sous-unités, comme les subordonnées par exemple, et ainsi augmenter la robustesse de nos modèles. Pour se convaincre de l'utilité de ces marques, il suffit de tester un système de dictée vocale grand public du marché en lui énonçant des enchaînements de phrases avec et sans ponctuation. Le résultat est meilleur avec des marques de ponctuation, car celles-ci sont parties intégrantes du modèle de langage.

Cet exemple illustre, comme l'indiquait déjà [Blanche-Benveniste et al. 90], que la langue parlée n'est pas directement dérivable du langage écrit. Des phénomènes, comme la variabilité de l'ordre des mots au sein du langage oral, introduisent une incertitude qui pose de nouveaux problèmes par rapport à l'écrit, structure beaucoup plus figée par la syntaxe et la grammaire. Les travaux de [Antoine et al. 01] ont montré qu'apparemment, ces inversions de mots ne sont pas corrélées avec la familiarité des locuteurs avec une tâche bien définie. Ceci tendrait à prouver, comme le remarquent les auteurs, qu'il existe un ensemble de constructions qui sont communes à tous les types de dialogues. Cela laisse penser qu'il y a là une certaine notion de généricité qui permettrait de limiter le travail de portage d'un système de dialogue adapté à un domaine vers un autre.

III.2. Objectifs

La discussion qui précède nous mène à fixer un objectif de recherche que l'on peut résumé ainsi : « améliorer la qualité de la reconnaissance de la parole, dans un cadre dialogique, en améliorant la modélisation statistique du langage oral spontané ». Concrètement, cela signifie que nous allons travailler à modéliser un langage sous la forme que nous avons décrite dans l'exemple I.1. Dans ce cas, l'univers du langage que doit intégrer la machine comporte de nombreuses formes propres au dialogue : des répétitions, des corrections, des inversions dans l'ordre des mots par rapport au langage écrit, etc.

Ces formes pourraient être catégorisées dans ce que nombre de chercheurs nomment extragrammaticalités. D'ailleurs, nous avons nous-même employé cette appellation dans diverses publications conformément à l'usage dans la littérature. Pourtant, en poussant plus loin la réflexion, cette appellation ne paraît pas cohérente avec ce qu'est intrinsèquement le langage oral. Ainsi, contrairement à l'expression écrite où tous ces événements sont des épi-phénomènes volontaires et purement rhétoriques, l'oral présente ces formes de manière récurrente. Il est évident que, dans l'éventualité où l'on voudrait posséder une grammaire pour le langage oral, les règles permettant la génération de tels évènements ne seraient pas des cas particuliers mais bien des règles générales. Dans ce cas, ces phénomènes deviennent grammaticaux. La notion d'extragrammaticalité est alors complètement caduque, puisqu'elle correspond à l'analyse d'énoncés oraux avec une grammaire représentant la syntaxe et les règles du français littéraire, ce qui équivaut à caractériser un ensemble d'éléments dans un espace avec la spécification d'un autre espace. Dans la suite de ce manuscrit, nous ne parlerons d'ailleurs plus d'extragrammaticalités, mais de spécificités du langage oral.

L'objectif de nos travaux de recherche est donc de fournir des outils et des méthodes pour faciliter la définition et l'obtention de modèles de langage gérant toutes ces spécificités de l'oral.

III.3. Choix d'une approche de modélisation

La difficulté de cette tâche de modélisation du langage oral tient à la nature même de ce dernier. Bien qu'il soit tout à fait possible de définir une grammaire pour le représenter, cela s'avère d'une très grande complexité, surtout lorsque le vocabulaire est de taille conséquente. De plus, la grammaire obtenue ne serait que l'expression d'un besoin ponctuel. Une nouvelle dépense d'efforts serait à fournir pour adapter cette grammaire à un autre domaine d'application.

Arrivé à ce point de notre raisonnement, il ne nous reste qu'une seule voie à explorer pour atteindre nos objectifs : la modélisation statistique du langage. L'avantage principal des méthodes probabilistes réside dans le fait qu'elles ne nécessitent pas l'intervention d'experts et qu'il est possible de les appliquer à d'autres domaines en changeant le corpus d'apprentissage. De plus, la modélisation statistique du langage a montré sa robustesse dans diverses langues, ce qui en fait l'outil le plus employé de nos jours [Rosenfeld 00], [Jelinek 01]. Cependant, pour l'utilisation de modèles statistiques, il est nécessaire de disposer de corpus textuels pour calculer des probabilités, et donc des modèles, de bonne

qualité. La qualité et la taille des corpus sont donc les principaux problèmes des approches statistiques. Cet état de fait est connu depuis longtemps et cela a débouché sur des expériences de collecte et de transcription de type magicien d'Oz. Dans ce cas, un humain, souvent dénommé le *compère*, remplace la machine et dialogue avec un autre humain. Les corpus audio sont ensuite transcrits manuellement. Comme le signale [Roussel 99], dans ces collectes, le problème provient principalement de l'humain et de sa capacité à communiquer. Il est très difficile pour un humain de simuler les erreurs d'un système de communication homme/machine. Ainsi se prive-t-on d'une bonne partie de la richesse qu'aurait pu fournir un corpus de ce genre. [Habert et al. 92] relevaient déjà des problèmes en termes de taille et de qualité de ces corpus :

« Pour l'oral, l'identification des classes à considérer est beaucoup moins avancée. [...] L'échantillon est trop petit pour bien représenter la population, ou l'échantillon est systématiquement biaisé et s'écarte significativement des caractéristiques de la population ».

L'idée conductrice de notre travail a donc été de trouver une source de textes permettant la modélisation du langage oral qui serait satisfaisante, d'un point de vue qualitatif et quantitatif, pour apprendre des modèles statistiques. Dans ce manuscrit, nous allons étudier ce que peut nous apporter, pour cette tâche, Internet, une forme d'écrit particulière de par sa multidisciplinarité et sa variété.

La chaîne complète, allant du signal de parole produit par l'utilisateur à la séquence de mots reconnue en passant par la description des algorithmes sous-jacents de la reconnaissance de la parole, fera l'objet du chapitre II. Le chapitre III présentera les modèles de langages les plus employés de nos jours.

Chapitre II : Reconnaissance de la parole

Chapitre II : Reconnaissance de la parole

Présentation du chapitre

Dans ce chapitre, nous nous intéresserons aux bases de la Reconnaissance Automatique de la Parole (RAP) et nous verrons quels sont les fondements théoriques des différents algorithmes utilisés. La présentation suivra la progression du signal de parole, partant de la production par l'être humain pour finir sous forme d'une chaîne de mots reconnue. Pour ce faire, nous détaillerons la façon dont l'ordinateur traite le signal de parole par le biais de sa paramétrisation. Nous verrons quelles sont les méthodes les plus employées actuellement pour la reconnaissance acoustique du signal. Nous terminerons par la présentation des algorithmes permettant la mise en correspondance des sons reconnus et de l'hypothèse finale du système de reconnaissance.

I. Principe général

La reconnaissance de la parole est fondée dans la plupart des systèmes actuels sur une approche probabiliste [Jelinek 01]. Les systèmes sont généralement constitués de deux unités principales, le module de décodage acoustico-phonétique et le module de modélisation du langage. Le schéma suivant présente les principales entités d'un système de reconnaissance.

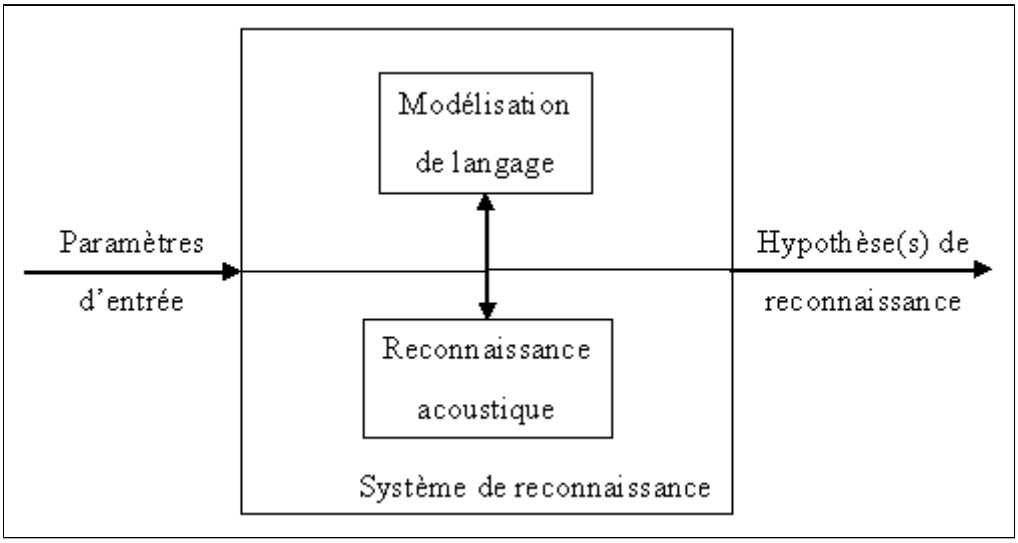


Figure II.1 : description symbolique d'un système de reconnaissance de la parole

Le premier module permet, à partir d'une analyse paramétrique du signal à reconnaître, de définir quel est l'élément acoustique qui est le plus probablement produit. Cet élément peut être de différents types : phonèmes, diphones, syllabes, etc. Cette étape franchie, il est nécessaire de mettre en correspondance une suite d'éléments acoustiques avec une forme lexicale. C'est ici qu'intervient le second module. Il permet d'obtenir une information *a priori* sur le positionnement d'un mot dans le signal à reconnaître par différentes techniques de modélisation soit à base de grammaire, soit purement statistique, soit à base d'approches mixtes telles que les grammaires probabilistes.

La formule générale, dans le cadre d'un système entièrement probabiliste, s'exprime sous la forme d'une équation bayésienne. Elle a été énoncée dans le cadre de la reconnaissance de la parole par [Bahl et al. 83]. Le but du système est de trouver l'hypothèse W^*qui maximise pour toutes les séquences de mots W possibles et pour une observation acoustique A, l'équation suivante :

$$W^* = \text{argmax}_W P(W|A) = \text{argmax}_W \frac{P(W).P(A|W)}{P(A)} \approx \text{argmax}_W P(W).P(A|W)$$

Équation II.1 : équation bayésienne de la reconnaissance de la parole

Dans cette équation, nous pouvons identifier plusieurs facteurs :

- P(A) est la probabilité de l'observation acoustique A. Celle-ci est constante pour toutes les séquences de mot W, d'où l'approximation finale de l'équation précédente. Pour générer cette observation, le module de décodage acoustique doit, dans un premier temps, analyser le signal de parole, et ensuite définir quelle est la suite d'éléments acoustiques la plus probable.
- P(A|W) est la probabilité de l'observation acoustique A connaissant une séquence de mots W. Pour ce faire, on utilise un dictionnaire phonétique c'est-à-dire contenant la transcription des graphèmes avec leurs décompositions en éléments de base pour le

système acoustique.

- P(*W*) est la probabilité *a priori* de la séquence de mots *W*, sans aucune notion d'acoustique, dans le langage considéré. C'est la probabilité générée par le modèle de langage. Les techniques de modélisation stochastique du langage seront exposées dans le chapitre suivant.

II. Du signal de parole à l'observation acoustique

II.1. Modules acoustiques

Les premiers modules de traitement dans un système de reconnaissance de la parole sont les suivants :

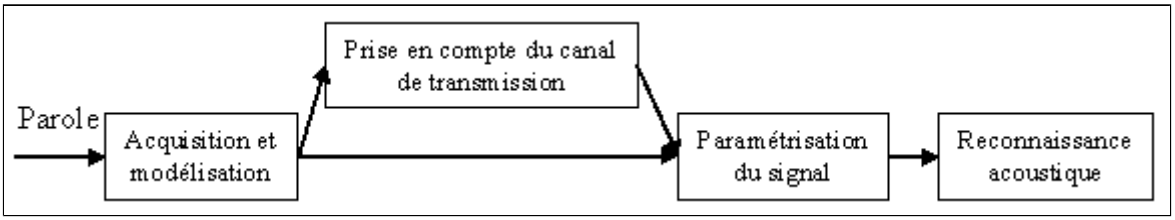


Figure II.2 : chaîne de traitement acoustique d'un système de reconnaissance de la parole

Comme le montre la figure II.2, le signal de parole est d'abord numérisé puis modélisé sous une forme généralement fréquentielle. Pourtant, avant d'obtenir ces mesures, le signal a subi des modifications dues à l'environnement dans lequel se trouve le locuteur, à l'influence du système d'acquisition, et à une éventuelle transmission par le biais d'un média informatique, par exemple un réseau. Ces modifications sont souvent regroupées sous le terme générique de « *canal de transmission* ». Certains systèmes de reconnaissance dispose d'un module de prise en compte de ce canal pour tenter d'éliminer son influence sur le signal de parole. Le module suivant, dans la chaîne de traitement acoustique, est celui qui extrait des paramètres pertinents pour la reconnaissance de la parole. Ces paramètres sont ensuite envoyés au module de reconnaissance acoustique qui identifie les sons présents dans le signal.

Détaillons chacun de ces modules pour comprendre l'enchaînement allant du signal de parole à l'observation acoustique. En ce qui concerne le module de reconnaissance acoustique, nous ne présenterons que la technique de reconnaissance la plus employée à l'heure actuelle : la modélisation par modèles de Markov. C'est celle que nous utilisons, nous aussi, dans notre système de reconnaissance de la parole. Nous n'aborderons pas la technique d'alignement temporel (*Dynamic Time Warping* en anglais) intégrée, par exemple, dans les téléphones portables. Le lecteur trouvera une explication détaillée de celle-ci dans [Web 09] et sa résolution par programmation dynamique dans [Sakoe et al. 78]. Nous ne présenterons pas non plus les approches fondées sur les réseaux de neurones ou sur des approches hybrides mélangeant modèles de Markov et réseaux de neurones. Le lecteur pourra trouver un état de l'art de ces techniques dans [Spalanzani 99].

II.2. Acquisition et modélisation du signal

II.2.a. Numérisation

Pour être utilisable par un ordinateur, un signal doit tout d'abord être numérisé. Cette opération tend à transformer un phénomène temporel analogique, le signal sonore dans notre cas, en une suite d'éléments discrets, les échantillons. Ceux-ci sont obtenus avec une carte spécialisée courante de nos jours dans les ordinateurs depuis l'avènement du multimédia. La numérisation sonore repose sur deux paramètres : la quantification et la fréquence d'échantillonnage.

La quantification définit le nombre de bits sur lesquels on veut réaliser la numérisation. Elle permet de mesurer l'amplitude de l'onde sonore à chaque pas de l'échantillonnage. De plus, cette quantification peut suivre une échelle linéaire ou logarithmique (comme l'échelle μ-law), cette dernière privilégiant la résolution de la quantification pour les niveaux faibles au détriment des niveaux forts.

Le choix de la fréquence d'échantillonnage est aussi déterminant pour la définition de la bande passante représentée dans le signal numérisé. Le théorème de Shannon [Bellanger 95] nous indique que la fréquence maximale f_{max} présente dans un signal échantillonné à une fréquence f_e est égale à la moitié de f_e. Un signal échantillonné à 16000 Hertz contient donc une bande de fréquences allant de 0 à 8000 Hertz. D'après ce principe, il est donc inutile de numériser un signal téléphonique à plus de 6800 Hertz, car le résultat ne contiendrait pas plus d'informations fréquentielles. Pourtant, comme la majorité des cartes ne proposent que certaines fréquences d'acquisition, le signal téléphonique est généralement échantillonné à une fréquence de 8000 Hz, ce qui, de plus, facilite la définition de filtres fréquentiels.

II.2.b. Transformée de Fourier

Joseph Fourier a montré que toute onde physique peut être représentée par une somme de fonctions trigonométriques appelée série de Fourier. Elle comporte un terme constant et des fonctions sinusoïdales d'amplitudes diverses. Ainsi un son sinusoïdal ne comporte qu'une seule raie spectrale correspondant à la fréquence de sa fonction sinus. Un son complexe est composé d'une multitude de ces raies spectrales qui représentent sa composition fréquentielle [Bellanger 95].

Dans le cas d'une séquence d'échantillons, il est alors possible de calculer une Transformée de Fourier Discrète (TFD, *Discret Fourier Transform* - DTF - en anglais). L'équation II.2 donne le calcul de la FFT pour une séquence X(n) comportant N échantillons.

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k)\, e^{-j\, k\, 2\, \pi\, (n/N)}$$

Équation II.2 : formule de la Transformée de Fourier Discrète

En 1965, [Cooley et al. 65] ont proposé un algorithme de calcul rapide de transformée de Fourier discrète, la *Fast Fourier Transform* (FFT, Transformée de Fourier Rapide - TFR - en français). La seule limitation de cet algorithme est que la taille de la séquence dont on veut obtenir la FFT doit être une puissance de 2. Le temps de calcul d'une FFT est environ 10 fois inférieur à celui d'une TFD classique. Le lecteur pourra trouver de plus amples informations à propos de ces algorithmes et des implémentations commentées dans [Press et al. 92].

II.3. Prise en compte du canal de transmission

Comme dans la tâche de reconnaissance de la parole, l'une des deux entités de la chaîne de communication est un ordinateur. De ce fait, il est nécessaire de prendre en compte le canal de transmission entre l'être humain et la machine, car celui-ci introduit des distorsions qui sont de nature à perturber suffisamment le signal de parole pour le rendre difficilement reconnaissable pour la machine. Ce canal de transmission est en général assimilé à un filtre. Il est possible d'inclure dans ce canal des informations comme la réponse impulsionnelle de la pièce où l'enregistrement est effectué, ou encore le bruit de fond.

Si l'on prend comme exemple l'enregistrement via un microphone, la réponse en fréquence de ce dernier introduit une distorsion qui modifie les fréquences identifiables dans le signal. Si l'enregistrement de la voix est réalisé par le biais d'une ligne téléphonique, la réduction fréquentielle est encore plus forte. En effet, dans ce cas, la bande passante se situe entre 300 et 3400 Hertz, ce qui élimine toutes les autres fréquences. De plus, avec l'arrivée des serveurs de reconnaissance distribuée [Klautau et al. 00], le canal peut aussi comporter une transmission via le réseau Internet. Dans ce cas, nous parlerons de transmission de Voix sur IP (*VoIP* pour *Voice over IP* en anglais) [Black 00]. Nous avons, nous aussi, décrit un protocole de transmission de données pour la reconnaissance et la synthèse de la parole via le réseau Internet [Vaufreydaz et al. 99a]. Les applications de visioconférence, entre autres, emploient de tels protocoles. Dans ce cas-là, le canal provoque non seulement une distorsion due au codage de la voix mais aussi, du fait que l'implémentation de ces protocoles est basée sur UDP/IP, une perte de paquets et donc de données dans le signal à reconnaître.

Il existe plusieurs façons de s'affranchir du canal par lequel le signal passe pour obtenir des résultats optimaux de reconnaissance. Il faut soit réduire la différence entre les données servant à apprendre les modèles de reconnaissance, soit réaliser un prétraitement pour annuler les effets du canal. Ces deux méthodes posent néanmoins des problèmes. La première méthode nécessite d'avoir une connaissance du canal et de pouvoir construire des bases acoustiques pour l'apprentissage des modèles acoustiques. La seconde, souvent basée sur des filtres adaptatifs, permet de s'adapter en cours de reconnaissance. Dans ce cas, il est nécessaire de connaître le type du canal. Le lecteur trouvera des clés pour comprendre les techniques d'adaptation dans [Sagayama 01].

II.4. Extraction de paramètres

Nous avons vu comment l'ordinateur appréhendait un signal sonore. Pourtant les formes temporelles ou fréquentielles ne sont pas les plus adéquates pour la reconnaissance de

la parole continue. Il est nécessaire de calculer plusieurs paramètres dérivés de ce signal. Nous n'aborderons ici que les principaux utilisés dans la littérature, et par nous-même dans notre système d'expérimentation.

II.4.a. Énergie du signal

Après la phase de numérisation et surtout de quantification, le paramètre intuitif pour caractériser le signal ainsi obtenu est l'énergie. Cette énergie correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames de signal successives pour pouvoir mettre en évidence des variations. La formule de calcul de ce paramètre est :

$$E(\textit{fenêtre}) = \sum_{n \in \textit{fenêtre}} |n|^2$$

Équation II.3 : calcul de l'énergie d'un signal échantillonné

Il existe des variantes de ce calcul. L'une des plus utilisées réalise une simple somme des valeurs absolues des amplitudes des échantillons pour alléger la charge de calcul, les variations restant les mêmes. D'autres, comme celle de [Taboada et al. 94] proposent la modification suivante du calcul intégrant une normalisation par rapport au bruit ambiant.

$$E(\textit{fenêtre}) = \log\left(\sum_{n \in \textit{fenêtre}} \frac{|n|^2}{R} \right)$$

Équation II.4 : calcul de l'énergie normalisé par rapport au bruit ambiant

Dans cette équation, *R* est la valeur moyenne de l'énergie du bruit. Le résultat de ce calcul tend vers 0 lorsque la portion considérée est une zone où il n'y a que le bruit de fond. Tout le problème de cette variante réside dans l'estimation du facteur de normalisation *R*.

II.4.b. Mel-scaled Frequency Cepstral Coefficients (MFCC)

Les travaux de Stevens [Stevens et al. 40] ont permis la mise en évidence de la *loi de puissance* ou *loi de Stevens* selon laquelle l'intensité de la perception d'un stimulus n'augmente pas linéairement en fonction de sa puissance mais de façon exponentielle en tenant aussi compte des modalités de l'expérimentation. Les coefficients MFCCs [Davis et al. 80] pour *Mel-scaled Frequency Cepstral Coefficients*, aussi nommés *Mel Frequency Cepstral Coefficients* dans la littérature, sont donc basés sur une échelle de perception appelée Mel, non linéaire. Celle-ci peut être définie par la relation suivante entre la fréquence en Hertz et sa correspondance en mels :

$$M_{mels} = x \cdot \log\left(1 + \frac{f_{Hz}}{y} \right)$$

Équation II.5 : correspondance entre l'échelle Mel et la fréquence en Hertz

Plusieurs valeurs sont utilisées pour *x* et *y*. En 1989, on trouvait dans [Calliope 89] *x = 1000/log(2)* et *y = 1000*. De nos jours, les valeurs les plus couramment utilisées sont

x = 2595 et *y = 700*. D'autres définitions de cette échelle peuvent être trouvées comme par exemple [Umesh et al. 99].

Pourtant l'utilisation de cette unité n'est pas suffisante. Pour avoir une largeur de bande relative qui reste constante, le banc de filtres Mel est construit à partir de filtres triangulaires positionnés uniformément sur l'échelle Mel donc non uniformément sur l'échelle fréquentielle. Cette répartition est illustrée ci-dessous :

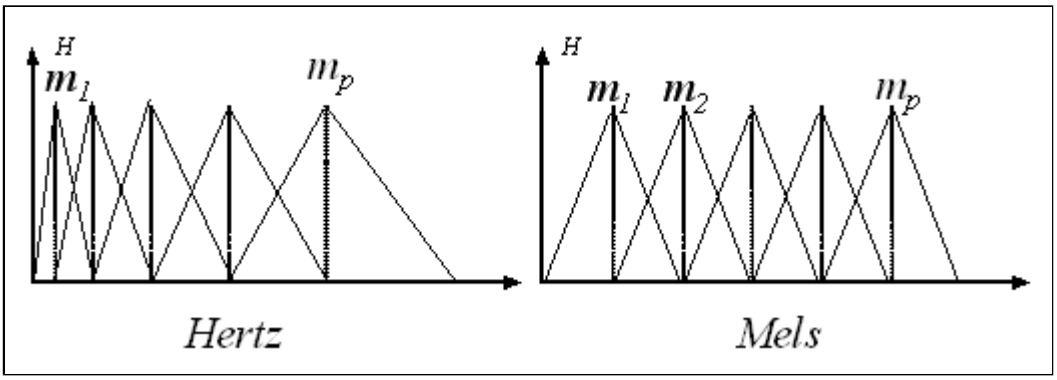


Figure II.3 : répartition des filtres triangulaires sur les échelles fréquentielle et Mel

Sur cette illustration, m_p correspond au nombre de filtres que l'on souhaite. Lorsque ce banc de filtres est en place, il est alors possible de calculer les coefficients MFCCs. L'algorithme peut être décrit comme suit :

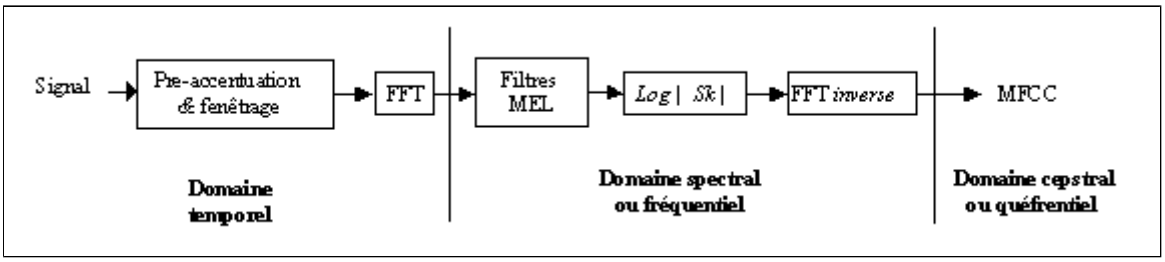


Figure II.4 : algorithme de calcul des MFCCs

Il est possible de choisir le nombre de paramètres générés en sortie de cet algorithme. Dans la littérature, le nombre de coefficients utilisés varie de 5 à plus d'une quarantaine en fonction de l'utilisation qui en est faite : reconnaissance de la parole, de la langue ou identification du locuteur par exemple. . En ce qui concerne le nombre de filtres, nombreux sont ceux qui choisissent 30 pour un signal avec une bande passante de 0 à 8 KHz.

II.4.c. Taux de passage par zéro

Le taux de passage par zéro (*zero crossing rate* en anglais) représente le nombre de fois que le signal, dans sa représentation amplitude/temps, passe par la valeur centrale de l'amplitude (généralement zéro). Il est fréquemment employé pour des algorithmes de détection de section voisée/non voisée dans un signal. En effet, du fait de sa nature aléatoire, le bruit possède généralement un taux de passage par zéro supérieur à celui des parties voisées.

Le comptage du nombre de passages par zéro est très simple à effectuer. Dans un premier temps, il faut enlever le décalage d'amplitude (*offset* en anglais), produit par la majorité des matériels d'acquisition, pour centrer le signal autour de zéro. Ensuite, pour chaque trame, il suffit de dénombrer tous les changements de signe du signal. Pour éliminer certains phénomènes parasites, [Taboada et al. 94] ont proposé une méthode nommée le *band-crossing*. Un seuil d'amplitude S permet de définir une zone autour du zéro de largeur 2xS au sein de laquelle les oscillations ne sont pas prises en compte. La formule du *band-crossing* pour chaque fenêtre analysée est donc :

$$\mathrm{BCR}(\textit{fenêtre}) = \sum_{n \in \textit{ fenêtre}} | f(n)-f(n-1) | \text{ avec } f(n) = \begin{cases} 1 \text{ si } n > S \\ f(n-1) \text{ si } -S \leq n \leq S \\ -1 \text{ si } n < -S \end{cases}$$

Équation II.6 : formule de calcul du Band-Crossing Rate

Cette mesure se montre très intéressante, dans le cadre d'une détection de parole en amont d'un système de reconnaissance, pour la détection de fricative en fin de signal à reconnaître ou d'attaque de plosive [Aubry et al. 00].

II.4.d. Autres paramétrisations du signal

Nous n'énumérerons pas tous les types de paramètres employés dans le domaine de la recherche en parole car il y en a énormément et ce n'est pas le propos de notre thèse. Pourtant, il est à noter que d'autres approches plus proches de l'audition humaine, telles les modèles d'oreille [Caelen 85], ont été étudiées. De plus, le lecteur trouvera des informations sur différents paramètres très largement utilisés dans [Rabiner et al. 93] pour le codage LPC (*Linear Predictive Coding*) présent dans la norme GSM, dans [Hermansky 90] pour les PLPs (*Perceptual Linear Predictive*) et [Hermansky et al. 94] pour les RASTA-PLP, version approfondie des PLP. Cette liste ne se veut pas exhaustive mais permet d'avoir un aperçu des différents paramètres qu'il est possible d'extraire d'un signal de parole.

II.4.e. Dérivées première et seconde

Le but final de l'extraction des paramètres est de modéliser la parole, un phénomène très variable. Par exemple, même si elle a de l'importance, la simple valeur de l'énergie n'est pas suffisante pour donner toute l'information portée par ce paramètre. Il est donc souvent nécessaire de recourir à des informations sur l'évolution dans le temps de ces paramètres. Pour cela, les dérivées première et seconde sont calculées pour représenter la variation ainsi que l'accélération de chacun des paramètres. Même si la robustesse de la représentation obtenue est accrue, cela implique aussi de multiplier par 3 l'espace de représentation.

II.4.f. Réduction de l'espace de représentation

Comme nous venons de le voir, l'espace de représentation du signal est souvent de taille conséquente, généralement de plusieurs dizaines de paramètres. Il est donc important de ne garder que des paramètres discriminants. La méthode majoritairement utilisée, de nos jours, est l'analyse discriminante linéaire [Siohan 95], LDA pour *Linear Discriminant Analysis* en

anglais. Cette technique s'apparente à l'analyse en composantes principales (ACP). Elle permet l'obtention de paramètres considérés comme discriminants en appliquant une transformation linéaire de l'espace d'entrée de taille n vers un espace de taille réduite q ($q < n$). L'application de cet algorithme maximise la séparation des classes qui sont affectées à chaque vecteur acoustique et ainsi améliore la robustesse de la représentation. [Haeb-Umback et al. 92] ont d'ailleurs montré que l'utilisation d'une telle analyse permet de pallier certaines catégories de bruits.

II.5. Reconnaissance acoustique par Modèles de Markov Cachés

Les Modèles de Markov Cachés (MMC), dont nous préférerons employer l'acronyme anglais HMM pour Hidden Markov Models, sont, à l'heure actuelle, les outils de modélisation les plus employés en reconnaissance de la parole continue.

II.5.a. Description

Les HMM ont montré leur adéquation à traiter la parole [Rabiner et al. 93]. Ce sont des automates stochastiques permettant de déterminer la probabilité d'une suite d'observations. Ils sont définis par l'ensemble de données suivantes :

- une matrice A qui permet la définition de la topologie du HMM en indiquant les probabilités de transition d'un état q_i vers un autre état (ou lui-même), soit $p(q_j| q_i) = a_{ij}$. Les modèles utilisés en reconnaissance de la parole sont d'ordre 1, c'est-à-dire que la probabilité de passer dans l'état suivant dépend uniquement de l'état courant. La taille de cette matrice est LxL, où L est le nombre d'états du modèle.
- une matrice B qui contient les probabilités d'émission des observations dans chaque état $b_j(x_n) = p(x_n| q_j)$. Dans le cas de la parole continue, cette probabilité est de type multigaussienne, définie uniquement par les vecteurs moyens, les matrices de covariance et des poids associés à chaque gaussienne.
- Une matrice Π donne la distribution de départ des états, c'est-à-dire pour chaque état la probabilité d'être atteint à partir de l'état initial q_I. Cet état est particulier puisqu'il ne peut émettre d'observations.

Le lecteur trouvera de plus amples informations sur les HMM dans [Rabiner et al. 89] et [Rabiner et al. 93]. La parole étant un phénomène temporel, l'utilisation de HMMs pose comme postulat que la parole est une suite d'évènements stationnaires. La topologie principalement employée dans la littérature est un modèle gauche-droit d'ordre 1 dit de Bakis.

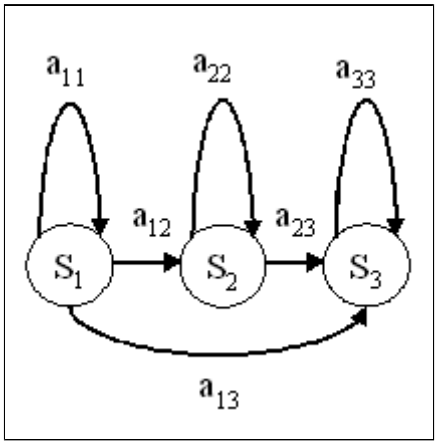


Figure II.5 : modèle HMM dit gauche-droit d'ordre 1 à 3 états

Cette topologie permet de prendre en compte les variations d'élocution dans le signal de parole. Ainsi, sur de la parole lente, comme nous l'avons vu pour la DTW, il y a répétition d'états, c'est pour cela que ce modèle de HMM permet de boucler sur un état (transitions a_{ii} sur la figure précédente). *A contrario*, si la parole est rapide, il est aussi possible de sauter l'état suivant (arc a_{13}). Cette topologie permet la modélisation des variations temporelles au sein du signal de parole.

Hormis l'élocution, un autre phénomène variable dans la parole provient des variations de prononciation de chaque locuteur, et des différences entre locuteurs. Il est aisément compréhensible que deux personnes, même si elles prononcent le même énoncé, n'ont pas exactement le même résultat acoustique. Cela s'accentue encore entre hommes et femmes.

La variation liée à un locuteur peut-être dûe à un état émotionnel particulier, le stress par exemple, ou à une altération temporaire de la voix suite à une maladie. C'est pour prendre en compte ces variations que les modèles HMMs utilisés de nos jours sont basés sur des fonctions multigaussiennes. Elles permettent de prendre en compte la variabilité autour de la moyenne calculée sur les données d'apprentissage.

Pour conclure, les HMMs et leurs caractéristiques représentent donc deux processus stochastiques distincts imbriqués :

- le premier est la suite d'observations produites $O = O_1\ O_2\ ...\ O_n$
- le second est la suite d'états parcourus $Q = q_1\ q_2\ ...\ q_n$

Les modèles de Markov sont dits cachés parce que la suite d'états parcourus pour générer la séquence O, n'est pas directement observable.

II.5.b. Modèles d'allophones

Les phonèmes représentent en phonologie le découpage des mots en sous-unités acoustiques. Pourtant, si l'on étudie par exemple le son « *a* », on observe qu'il est légèrement différent en fonction des phonèmes qui le précèdent et qui le suivent : c'est le phénomène de coarticulation. Il est alors possible de créer un HMM différent pour chaque « *a* » en contexte. La majorité des systèmes tiennent compte du contexte gauche et droit, ce qui aboutit à une modélisation dite par *triphones*. Dans ce cas, le nombre de représentants dans le corpus

acoustique, pour l'apprentissage des modèles, peut devenir insuffisant. Il est alors possible, en regroupant les phonèmes des contextes gauche et droit en classes, d'obtenir des modèles plus génériques dépendant du contexte.

Les classes peuvent être de différents types mais sont très souvent de nature acoustique. Nous citerons, à titre d'exemple, les plosives, les fricatives, les liquides, les voisées et les non voisées mais il en existe une multitude d'autres. En général, l'algorithme de classification est de type K-Moyennes (*KMeans*) ou à base d'arbre de décision [Rabiner et al. 93] et permet l'obtention, en fonction du nombre de représentants de chaque classe dans le corpus d'apprentissage, du nombre optimal de classes que l'on peut apprendre.

Ces classes sont nommées allophones et permettent d'affiner la modélisation des phonèmes. Cependant, elles apportent un surcoût de mémoire, pour leur stockage lors de l'utilisation des HMMs, mais aussi de temps processeur car le nombre de modèles à évaluer pendant la phase de reconnaissance augmente très rapidement.

II.5.c. Problème de l'apprentissage

L'un des principaux problèmes de l'utilisation des HMMs réside dans la phase d'apprentissage, qui conduit à l'évaluation de tous les paramètres du modèle. Il s'agit avec un corpus d'apprentissage, contenant un étiquetage par sous-unités acoustiques du signal temporel, de maximiser la vraisemblance que le modèle HMM ait produit la suite d'observations. Il existe plusieurs algorithmes pour faire cela : l'algorithme dit de *Baum-Welch* [Baum et al. 66], le *forward-backward* [Baum 72] ou même simplement à l'aide de l'algorithme *Viterbi* [Forney 73] comme l'indique [Deroo 98] page 46. Cette problématique d'apprentissage des modèles acoustiques n'étant pas directement reliée à notre thèse, nous ne détaillerons pas ces algorithmes dans ce manuscrit. Pour de plus amples informations sur ces algorithmes, le lecteur pourra se référer à [Rabiner et al. 89], [Rabiner et al. 93] et [Deroo 98].

III. De l'observation acoustique à la forme lexicale finale

Nous avons vu comment apprendre à reconnaître des formes phonétiques de base. Il reste maintenant à déterminer la forme lexicale finale correspondante.

III.1. Dictionnaire phonétique et modèles d'unités plus longues

Dans le cadre de la reconnaissance de la parole continue, même si le système acoustique est basé sur des phonèmes, il faut obtenir, pour chaque entrée du dictionnaire phonétique, un modèle qui lui est propre. Ces modèles sont obtenus par concaténation de HMMs de phonèmes.

Dans notre système, le dictionnaire phonétique est sous la forme suivante :

{a}	{a}
{a(2)}	{{h WB} {a WB}}
{absolument}	{{a WB} b s O l y m {an WB}}
{absolument(2)}	{{a WB} b s O l y m an {t WB}}
{accès}	{{a WB} k s {E WB}}
...	...

Exemple II.1 : extrait d'un dictionnaire acoustique

Plusieurs points sont remarquables dans cet exemple. Premièrement, il existe des variantes phonétiques pour chaque entrée lexicale. Cela permet non seulement de mieux couvrir les variantes locales de prononciation du français par exemple, mais aussi de gérer les phénomènes de liaisons entre les mots. Si l'on regarde la variante du mot "absolument" qui finit par « *t* » (seconde variante, ligne 4), elle permet de modéliser la liaison, dans un contexte comme "absolument il", par exemple, pour le cas où une personne la réaliserait. Ensuite, les marqueurs de début et de fin de mots (WB pour *Word Boundary*) sont indiqués clairement pour les entrées de plus d'un phonème. À l'aide de ce dictionnaire et des HMMs de chaque phonème, il est possible de construire, par concaténation, un ou plusieurs modèles pour chaque mot.

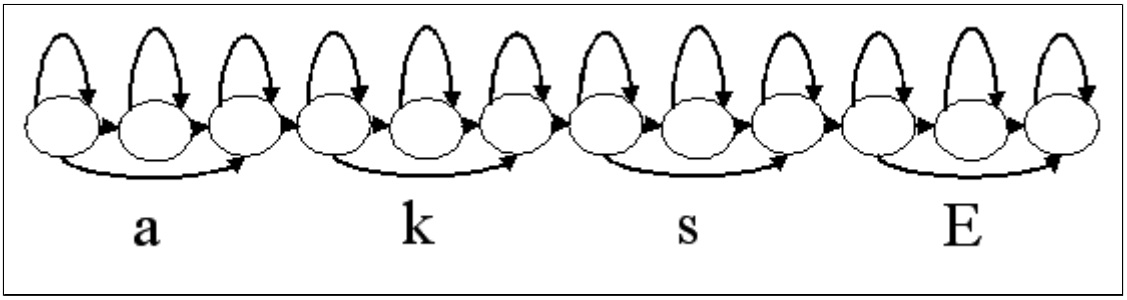


Figure II.6 : HMM du mot accès obtenu par concaténation de HMMs de phonèmes

Bien entendu, dans l'éventualité de l'utilisation d'allophones, cette concaténation tient compte des contextes des phonèmes. Il est possible d'obtenir des modèles de phrases en concaténant à leur tour des modèles de mots. Cela est particulièrement intéressant dans les procédures d'alignement de phonèmes sur des signaux dont on possède la transcription phonétique.

III.2. Algorithmes de recherche

De nos jours, il existe beaucoup d'algorithmes de reconnaissance basés sur des HMMs. Il existe presque autant de variantes que de systèmes de reconnaissance. Nous ne présenterons ici qu'un état de l'art des principaux algorithmes des systèmes de reconnaissance.

III.2.a. Généralités

Avec les progrès intervenus au cours de la dernière décennie, les capacités des systèmes de reconnaissance ont considérablement augmenté. Elles sont passées de quelques mots reconnus en mode isolé pour un seul locuteur à des systèmes multilocuteurs avec plusieurs milliers de mots, en parole continue voire spontanée. Tout cela amène de nouveaux problèmes. Il est nécessaire de limiter l'espace de recherche, qui croit de manière exponentielle, pour obtenir le bon résultat avec un temps de traitement correct. Les algorithmes de reconnaissance incluent donc très souvent des statégies permettant de choisir un nombre limité d'hypothèses à chaque instant, et ainsi de n'explorer que l'espace suffisant pour trouver la meilleure solution. Les travaux de [Woszczyna 98] montrent que de nombreuses stratégies peuvent être intégrées à différents niveaux des systèmes de reconnaissance, sans pour autant dégrader les résultats.

III.2.b. Algorithme A^* ou *A étoile*

L'algorithme A^* a été adapté de nombreuses fois pour être intégré dans des systèmes de reconnaissance de la parole. Considérons un graphe G défini par deux ensembles, S contenant l'ensemble des sommets et A celui des arcs, un arc étant un couple (a_n,a_m) de sommets reliés entre eux. Dans notre application, le graphe représente les différentes possibilités de progression sur le chemin acoustique. La particularité de l'algorithme A^* est d'utiliser une fonction heuristique pour guider la recherche qui dépend non seulement du chemin déjà parcouru mais aussi d'une estimation du chemin qui reste à parcourir. Cette fonction peut s'exprimer avec des probabilités exprimées sous forme logarithmique comme dans l'équation II.7.

$$f(n) = g(n) + h(n)$$

Équation II.7 : fonction heuristique pour l'algorithme A^*

Dans cette équation, nous trouvons $g(n)$ qui représente le score du chemin pour arriver à l'état courant n et $h(n)$, une estimation du score pour atteindre le nœud final. Cette deuxième fonction est souvent appelée *sonde*, puisqu'elle permet de guider l'algorithme pour qu'il choisisse le chemin le plus prometteur. Il est aisé de comprendre que le problème de l'utilisation du A^* se résume à la définition d'une fonction h exprimant correctement le poids du chemin jusqu'au nœud final. Plusieurs approches ont été proposées comme notamment, l'utilisation pour g et h des fonctions de l'algorithme forward-backward employées lors de la phase d'apprentissage du module acoustique [Kenny et al. 91]. À chaque pas de l'algorithme, pour tous les chemins en cours, le chemin le plus prometteur au sens de f est étendu, l'hypothèse de reconnaissance étant le chemin qui atteint le nœud final.

L'avantage principal de l'algorithme A^* est de pouvoir fournir en une seule passe les n meilleurs chemins au sein du graphe, il suffit pour cela de garder une pile des n chemins les plus prometteurs et de les étendre à chaque fois, et ainsi de limiter l'espace de recherche tout en fournissant plusieurs résultats [Soong 91].

III.2.c. Algorithme à base de modélisation arborescente

Si l'on examine un dictionnaire phonétique, on se rend vite compte qu'il existe beaucoup de préfixes communs à tous les mots du dictionnaire. Une modélisation arborescente devient alors très efficace pour représenter les suites de phonèmes des mots, comme elle l'est pour représenter des graphèmes. Elle permet un gain de place non négligeable en mémoire, mais introduit une incertitude au niveau des algorithmes de reconnaissance. En effet, dans une représentation standard des mots, il est facile de connaître instantanément le mot en cours lors de la phase de reconnaissance, alors que cela devient impossible dans une représentation arborescente tant que l'on n'a pas atteint une feuille de l'arbre. L'exemple ci-dessous illustre la représentation arborescente avec l'extrait de dictionnaire phonétique donné précédemment :

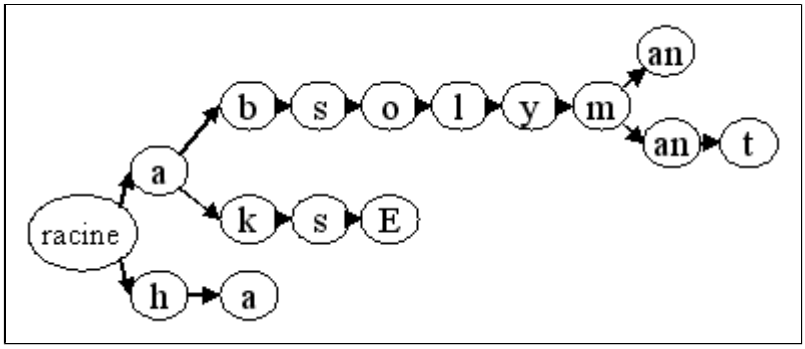


Figure II.7 : représentation arborescente d'un dictionnaire phonétique

Ce mode de stockage est aussi très intéressant, comme nous pouvons le voir sur la figure, dans le cas de variantes phonétiques de mêmes mots. Certaines approches ont aussi utilisé la mise en commun des suffixes, ce qui aboutit à la construction d'un graphe. En utilisant ces méthodes, dans un système reconnaissant 65000 mots basés sur des phonèmes simples, il ne subsiste, au début de chaque nouveau mot, que l'ensemble des phonèmes connus à évaluer, puisque tous les débuts de mots ont été factorisés [Gopalakrishnan et al. 95]. Même dans le cas d'un système utilisant des allophones, cela entraîne un gain non négligeable en temps de reconnaissance puisque le maximum de HMMs à considérer est le nombre d'allophones présent dans le système.

Pourtant, la non-connaissance du mot en cours introduit un problème dans l'intégration du modèle de langage. Ainsi, lorsque l'on arrive à la fin d'un mot (feuille de l'arbre), il faut choisir le meilleur chemin, soit le meilleur prédécesseur de ce mot, en utilisant le modèle de langage. Dans certains cas, cette technique peut conduire à des erreurs, l'information du modèle de langage étant incluse trop tard, en fixant la meilleure position pour le début du mot courant en se basant uniquement sur les scores acoustiques de l'hypothèse en cours. [Woszczyna 98] a montré que cela pouvait conduire à des problèmes de segmentation et de décision qui conduisent à un accroissement des erreurs pouvant aller jusqu'à 12%. Pour pallier ce problème, [Ney et al. 92] proposent de garder, pour tous les prédécesseurs possibles une copie de l'arbre. Pourtant, si l'on ne limite pas les hypothèses, ces copies peuvent rapidement prendre beaucoup de mémoire. On peut employer alors une méthode d'élagage d'arbre (*pruning* en anglais) nommée *Beam search*. Plutôt que d'utiliser un seuil

fixe pour réduire l'espace de recherche en éliminant les hypothèses peu probables, cette technique utilise comme référence le score de la meilleure hypothèse. Un seuil empirique est alors fixé pour définir quels sont les chemins, par rapport au plus prometteur, qui seront gardés. Ainsi, il est possible de limiter le nombre de copies des arbres que l'on possède en mémoire. Cette technique possède aussi l'énorme avantage, par rapport à un seuil fixe, de ne jamais conduire à des non réponses car on garde toujours au moins la meilleure hypothèse.

III.2.d. Algorithme de résolution de treillis de mots

Les algorithmes résolvant des treillis de mots sont très souvent utilisés comme seconde passe de reconnaissance. Il s'agit d'utiliser les résultats d'une précédente phase de reconnaissance afin de construire un graphe de mots réévalué avec des méthodes plus coûteuses, en temps de calcul ou en mémoire, pour trouver une solution optimale, comme par exemple un modèle acoustique présentant plus de modèles d'allophones, un modèle de langage différent, etc.

La construction de ce graphe est basée sur la pile des résultats de la phase de reconnaissance précédente. Ainsi, tous les mots finissant au début d'un autre mot sont ses prédécesseurs dans le graphe. Il est possible aussi d'inclure, sur chaque nœud, la probabilité de l'hypothèse ayant produit le chemin menant à ce mot. Un exemple de graphe est donné dans la figure suivante.

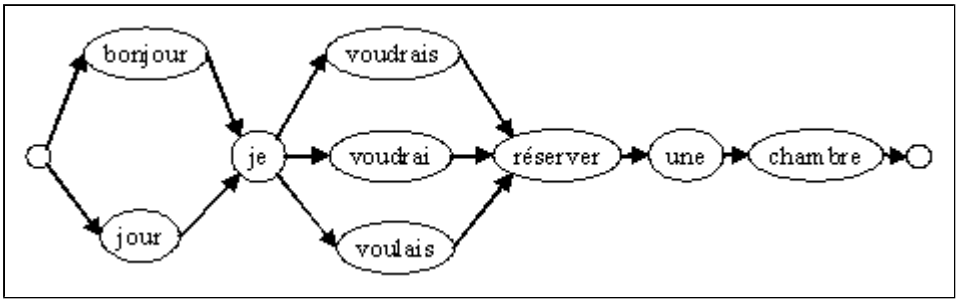


Figure II.8 : Treillis de mots

Sur cet exemple, on note que le chemin final est commun à toutes les hypothèses. Par contre, au début de la phrase, il y a eu un problème pour la reconnaissance du préfixe *bon* du mot *bonjour*. Enfin, il y a confusion entre différentes formes fléchies du verbe *vouloir*. Si l'on applique au treillis précédent un modèle de langage trigramme dédié à la réservation touristique, l'hypothèse la plus probable trouvée par le système est "bonjour je voudrais réserver une chambre". On peut voir que, sur des critères purement linguistiques, la solution la plus probable est le conditionnel et non le futur ou l'imparfait.

Ces treillis de mots peuvent être aussi considérés comme la sortie du système de reconnaissance. Cela s'avère intéressant, par exemple, dans des applications de traduction automatique de parole ou de dialogue homme/machine pour tenter de corriger les erreurs commises par le système de reconnaissance [McNair et al. 94].

Conclusion

Dans ce chapitre, nous avons décrit les premiers modules du traitement de la parole en vue de sa reconnaissance. Nous avons abordé les outils et les algorithmes les plus répandus de nos jours. Nous avons présenté ces données pour caractériser notre système d'expérimentation, décrit en détail plus loin dans ce manuscrit. Nous avons jusqu'ici volontairement éludé les modèles de langage. Ceux-ci étant la base de notre travail, le chapitre suivant leur sera entièrement consacré.

Chapitre III : Modélisation statistique du langage

Chapitre III : Modélisation statistique du langage

Présentation du chapitre

Avant de chercher à améliorer les techniques de modélisation statistique du langage, il convient de les présenter. Nous commencerons par les modèles n-grammes et n-classes et leurs principales variantes. Nous aborderons ensuite le problème majeur de l'utilisation des modèles statistiques, c'est-à-dire les séquences de probabilité nulle qui n'ont pas été rencontrées lors de l'apprentissage du modèle. Nous expliquerons quels sont les outils et les méthodes qui permettent de remédier à ce problème.

I. Modèles probabilistes

I.1. Généralités

Les modèles de langage probabilistes sont très répandus dans de nombreuses applications. Ils sont majoritairement employés, seuls ou avec à d'autres techniques, pour la reconnaissance de caractères manuscrits [Srihari et al. 92], en traduction de parole [Brown et al. 90], [Cattoni et al. 01], et bien entendu en reconnaissance de la parole [Rosenfeld 00].

Dans un système de reconnaissance de la parole, comme nous l'avons vu dans les précédents chapitres, on utilise la seule connaissance linguistique pour déterminer la suite de mots énoncés dans le signal à reconnaître, en fonction des sous-unités trouvées par le module acoustique. Les systèmes à base de modèles de langage statistiques tentent de déterminer la probabilité *a priori* de la séquence de mots $S = m_1, m_2, ..., m_i$ selon l'équation III.1.

$$P(S) = P(m_1) \times P(m_2 \mid m_1) \times \ldots \times P(m_i \mid m_1, m_2, \ldots, m_{i-1})$$

Équation III.1 : formule générale pour le calcul du score d'une séquence de mots avec un modèle probabiliste

Le principal avantage de ces techniques par rapport à des modélisations à base de grammaires est le type de réponse proposée. Les grammaires engendrent généralement une réponse de type «oui/non», même s'il est aussi possible d'obtenir une série de réponses ordonnées. L'intégration de probabilités, au sein d'un système de reconnaissance de la parole, permet nombre de manipulations et de combinaisons qu'une simple réponse «oui» n'autorise pas. *A contrario*, le désavantage principal d'un modèle de langage probabiliste est sa capacité à générer, dans certaines conditions, des suites de mots complètement incohérentes avec une forte probabilité.

Il existe différentes techniques pour le calcul des probabilités d'un modèle de langage. Celles-ci sont généralement estimées sur des corpus d'apprentissage censés représenter au mieux le langage à modéliser. Il existe de nombreuses variantes, qui utilisent différentes informations, depuis le simple graphème jusqu'à des classes ou des séquences de mots.

I.2. Modèles n-grammes

I.2.a. Présentation

Le principal problème dans l'utilisation de modèles de langage probabilistes tient en la longueur de l'historique considéré. Il est effectivement très difficile de calculer efficacement la probabilité $P(m_n|m_1, m_2, \ldots, m_{n-1})$ au cours de la reconnaissance. De plus, l'estimation des probabilités du modèle est impossible car il n'existe généralement pas de corpus d'apprentissage comprenant tous les historiques possibles pour tous les mots. On approche alors la probabilité en fonction d'un historique de taille réduite et fixe. C'est ce que l'on nomme un modèle n-gramme. Le calcul considère alors, pour la prédiction d'un mot, que la suite des n-1 mots qui le précèdent est suffisante. Les termes de l'équation précédente se résument alors à :

$$P(m_i \mid m_1, m_2, \ldots, m_{i-1}) = P(m_i \mid m_{i-n+1}, \ldots, m_{i-1})$$

Équation III.2 : calcul du score d'une séquence de mots avec un modèle n-gramme

Ce type de modèle est actuellement le plus utilisé pour les systèmes de reconnaissance de la parole, principalement avec des historiques de longueur 2, ce qui correspond à des modèles dits trigrammes [Woszczyna 98], [Johnson et al. 99], [Gauvain et al. 00a], [Bacchiani et al. 01], [Jelinek 01]. Ces modèles se sont révélés très performants pour modéliser les différentes langues européennes. Le calcul de base de ces probabilités se fait donc par un comptage de chaque séquence observée. À partir de maintenant, nous exprimerons les formules en notant $h_n(m)$ l'historique de longueur n-1 du mot m. La longueur de cet historique dépendra de l'ordre des modèles. Par commodité, nous noterons simplement h pour un historique de longueur variable. Soit N la fonction qui, pour une séquence de mots,

donne le nombre de fois où cette séquence a pu être observée dans le corpus d'apprentissage. Le calcul de la probabilité d'apparition du mot m en fonction de son historique *h* s'exprime alors sous la forme :

$$P(m \mid h) = \frac{N(h, m)}{N(h)}$$

Équation III.3 : calcul de la probabilité de la séquence de mots h, m

Cette estimation devient difficile lorsque le vocabulaire croît. Ainsi, si l'on considère un modèle de langage basé sur un vocabulaire *V* comprenant |*V*| entrées lexicales, le nombre maximum de trigrammes de ce modèle est $|V|^3$. Dans l'éventualité d'un vocabulaire important, |*V*| > 1000 par exemple, le nombre maximum de trigrammes dépasse le million. Il est bien sûr évident qu'au sein du langage, même naturel, il n'existe pas toutes les combinaisons de mots. Pourtant, il n'est pas possible de garantir que tous les trigrammes possibles sont présents dans le corpus d'apprentissage comme nous l'avons déjà indiqué dans le chapitre I.

I.2.b. Variantes des modèles n-grammes

I.2.b.1. N-grammes distants

Dans l'approche standard, un mot est prédit par les mots qui le précèdent immédiatement. Pourtant, dans une langue comme le français, cela n'est pas toujours optimum dans le cas de certains mots. Par exemple, les adverbes temporels n'apportent souvent que peu d'informations sur le mot suivant. Les modèles dits distants se basent sur cette constatation et se proposent de prédire un mot en fonction d'un historique qui ne le précède pas immédiatement [Langlois et al. 00]. Un modèle trigramme distant d'ordre 2 utilise comme historique du mot m_i, les mots m_{i-3} et m_{i-2}.

L'utilisation de ces modèles distants apporte des améliorations lorsqu'ils sont combinés avec d'autres types de modèles. [Huang et al. 93] et [Langlois et al. 00] ont montré un gain de perplexité entre une combinaison utilisant des modèles distants de différents ordres et un modèle n-gramme simple.

I.2.b.2. Modèles cache et trigger

Les modèles cache [Kuhn et al. 90] et *trigger* [Lau 93] sont basés sur des relations à long terme entre les mots du langage considéré. Dans un document, ou une discussion dans le cas d'un dialogue, l'apparition de certains mots peut influencer la probabilité d'apparition de ce même mot, pour le modèle cache, ou d'un autre mot, pour le modèle *trigger*.

En 1992, des chercheurs d'IBM ont conduit des expérimentations basées sur le jeu de Shannon [Shannon 51]. Ce jeu avait pour but de mesurer la quantité d'information présente dans la langue anglaise. Pour cela, il était demandé à des sujets de prédire la lettre qui allait apparaître dans un texte, connaissant toutes les lettres apparues. L'adaptation de ce jeu, utilisée par IBM, consiste à faire prédire par des sujets le mot suivant dans un document en

connaissant tous les mots précédemment apparus dans le document. En comparant la capacité de prédiction d'un humain à celle d'un modèle statistique, il s'est avéré que, dans 40% des cas où le sujet était meilleur que le modèle statistique, le mot à prédire était présent dans l'historique.

Nous pouvons aussi citer les exemples fournis dans [Rosenfeld 94] page 18 à 20. D'après l'auteur, en anglais, la probabilité du mot *winter* (hiver) varie en fonction du nombre de fois où l'on retrouve le mot *summer* (été) dans l'historique du document. Cette probabilité se trouve ainsi très fortement augmentée dans le cas où l'on retrouve plus de 4 fois le mot *summer* précédemment.

Ces deux exemples montrent bien l'intérêt des modèles cache et *trigger*. L'avantage de ces deux modèles est la possibilité d'obtenir, par une simple interpolation linéaire d'un modèle trigramme avec des modèles cache et *trigger*, un modèle intégrant les connaissances de ces 3 sources selon l'équation suivante :

$$P(m \mid h) = \alpha.P_{\text{n-gram}}(m \mid h) + \beta.P_{\text{cache}}(m \mid h) + \delta.P_{\textit{trigger}}(m \mid h)$$

Équation III.4 : formule d'interpolation d'un modèle n-gramme avec des modèles cache et trigger

Dans l'équation III.4, les facteurs d'interpolation α, βet δ sont tels qu'ils somment à 1. Cette technique permet donc d'intégrer des contraintes locales, par le biais du modèle trigramme, mais aussi des informations à plus longue distance grâce au modèle *trigger* et au modèle cache.

I.2.b.3. Autres variantes

Il existe de très nombreuses autres variantes des modèles n-grammes. Le modèle permugramme [Schukat-Talamazzini et al. 95] fonctionne sur la base d'une interpolation linéaire de plusieurs niveaux de modèles n-grammes, en intégrant des variations dans l'ordre des mots de l'historique et cela pour prendre en compte la variabilité dans l'ordre des mots au sein du langage [Antoine et al. 01]. Le modèle x-gramme [Bonafonte et al. 96] se fonde sur la constatation que certains mots ne sont pas très utiles pour la prédiction des mots suivants. On déjà abordé plus haut le cas d'adverbes de temps comme "aujourd'hui" par exemple. Le principe est alors d'extraire dans l'historique de taille *n*-1, des mots jugés pertinents qui seront employés pour la prédiction. Enfin, le lecteur pourra s'intéresser à des techniques de fusion de différents modèles basées sur le principe du maximum d'entropie [Rosenfeld 94] ou aux modèles de langage structurels [Zitouni 00] chapitre 4.

I.3. Modèles n-classes

En raison du problème de manque de données d'apprentissage, il est nécessaire de trouver une méthode maximisant la quantité d'information utile. Pour cela, il est possible de regrouper les mots en classes. En effet, si l'on prend l'exemple d'un nom commun masculin singulier, sa probabilité d'apparaître après la préposition *un* est la même que celle d'un autre nom commun identique en genre et en nombre. Cependant, ces classes ne sont pas forcément de type syntaxique (nom commun, verbe, préposition, etc.), elles peuvent être d'autre nature,

par exemple des classes obtenues avec des méthodes de classification automatique [Kneser et al. 93b]. Les modèles obtenus avec cette approche se nomment modèles n-classes.

En appliquant ce principe de classification, le modèle prédit non plus un mot en fonction des *n*-1 mots le précédant, mais une classe de mot en fonction des *n*-1 classes qui la précèdent. En posant C_i la classe courante, l'équation du modèle trigramme se transforme, pour un modèle triclasse, pour donner :

$$P(C_i \mid C_{i-2}C_{i-1}) = \frac{N(C_{i-2}C_{i-1}C_i)}{N(C_{i-2}C_{i-1})}$$

Équation III.5 : calcul de la probabilité d'une classe dans un modèle n-classes

Le principal avantage de cette méthode réside dans le fait qu'un mot d'une classe donnée, ne se trouvant pas forcement dans le corpus d'apprentissage, hérite de la probabilité de tous les autres représentants de sa classe au sein de ce même corpus. Ainsi, il est théoriquement possible d'ajouter un mot dans le système sans pour autant avoir à réestimer la probabilité spécialement pour lui. C'est un avantage pour le changement d'échelle d'un modèle de langage puisqu'il n'est pas nécessaire de refaire tout le travail à chaque fois.

Le modèle, tel qu'il est présenté dans l'équation III.5, considère que tous les mots d'une même classe sont équiprobables. Dans le cas de classes syntaxiques, il apparaît que pour la classe des noms communs, cela n'est pas le cas : par exemple *maison* sera plus probable que *scrabble*. De plus, il est probable que certains mots appartiennent à différentes classes. Ainsi, dans le cas de classes syntaxiques, la forme lexicale *cours* peut être un nom commun masculin pluriel (des cours d'anglais par exemple), féminin pluriel (pour les cours d'une école) ou une forme fléchie du verbe courir. Les modèles n-classes actuels intègrent aussi la probabilité de chacun des mots au sein de leur classe. Il faut avoir un corpus contenant un étiquetage de chaque mot sous forme de classe. Pour cela, il est nécessaire de faire appel à des experts ou à des techniques de lemmatisation automatique [Schmid 94]. Par la suite, pour chaque mot et chacune de ses classes, la probabilité est calculée comme suit, en notant C(m), la fonction qui, pour un mot m, renvoie sa classe :

$$P(m \mid C(m)) = \frac{N(m)}{N(C(m))}$$

Équation III.6 : probabilité d'appartenance d'un mot à une classe

La probabilité d'un mot au sein d'une séquence est alors obtenue par la formule :

$$P(m \mid h) = P(m \mid C(m)) \times P(C(m) \mid h(C(m)))$$

Équation III.7 : calcul de la probabilité d'un mot dans un modèle n-classes

Cette formule réintroduit le problème de corpus que se proposait de résoudre cette modélisation basée sur des classes. De nouveau, il faut posséder un corpus étiqueté avec des classes de taille suffisante pour réaliser l'apprentissage. L'avenir de telles approches se situe

alors plus dans des techniques de classification automatique que dans des étiquetages manuels de classes de mots.

I.3.a. Variantes des modèles n-classes

Dans le cas de certains mots, il est très difficile de déduire à coup sûr quelle est la classe d'un mot en fonction d'un historique de taille réduite. Pour cela, les modèles POS [Cerf-Danon et al. 91] (*Part Of Speech*), utilisent, pour le calcul de score d'apparition d'un mot, la moyenne des probabilités conditionnelles d'apparition de ce mot dans cette position, dans chacune de ses classes d'appartenance. Les modèles morphologiques [El-Bèze et al. 90] sont une extension de ce modèle POS. Ils incluent aussi les probabilités de la suite des lemmes, déduits des mots, dans la séquence.

II. Problème du manque de données d'apprentissage

II.1. Énoncé du problème

Le problème majeur dans l'utilisation de modèles de langage statistiques est l'éventualité que celui-ci produise une probabilité nulle [Witten et al. 91] pour une séquence de mots qui n'a pas été rencontrée lors de l'apprentissage mais qui existerait tout de même dans le langage visé par l'application. Cela peut engendrer des problèmes lors de la phase de recherche des algorithmes de reconnaissance. Ainsi, si la suite de mots est parfaitement reconnue au niveau acoustique mais qu'elle a une probabilité nulle dans le modèle de langage, alors toutes les possibilités trouvées jusque là seront élaguées et le système ne fournira aucune réponse. Pour éviter cela, il est nécessaire de trouver un moyen permettant l'obtention d'une probabilité pour les événements inconnus.

Si l'on étudie de plus près un corpus d'apprentissage, ont peut noter que certains événements peuvent apparaître mais ne pas être représentatifs. Si par exemple, un événement est identifié une seule fois sur un ensemble de 100 millions d'événements, sa probabilité est alors vraiment infime. Cependant, même si cette probabilité très faible reflète ce qui a été observé dans le corpus, elle reste très supérieure à ce que cet événement porte réellement comme information. Il est alors raisonnable de ne pas prendre en compte cet événement. Sa probabilité, partie intégrante de la distribution de probabilité, peut être réutilisée pour donner une chance d'apparaître aux séquences de mots non vues en apprentissage. Cette constatation a amené les chercheurs à mettre au point des techniques de lissage et de redistribution de probabilités pour résoudre ce problème.

II.2. *Good-Turing discounting* et approche de Katz

En 1987, [Katz 87] proposa une approche basée sur les travaux de Good [Good 53]. Ce dernier avait mis au point une méthode de lissage des probabilités des événements observés au sein d'un corpus. Cette formule se nomme formule de Good-Turing. Notons n_r, le nombre d'événements qui apparaissent r fois dans le corpus d'apprentissage. Les travaux de Good ont montré qu'il est possible, pour récupérer un peu de la masse de probabilité totale, de redéfinir le compte des événements apparus r fois comme dans l'équation III.8.

$$\text{décompte}(\, r \,) = (\, r+1 \,)\, \frac{n_{r+1}}{n_r}$$

Équation III.8 : formule de Good-Turing

Alors la probabilité pour les mots non rencontrés avec comme historique h est définie dans l'équation III.9 :

$$\lambda(\, h \,) = 1 - \sum_{\text{m} \mid N(\, h, \text{m} \,) > 0} \left(\frac{\text{décompte}(\, N(\, h\ \text{m} \,) \,)}{N(\, h \,)} \right)$$

Équation III.9 : probabilité des séquences non rencontrées avec un historique donné

L'approche de Katz se propose d'utiliser ces formules pour pouvoir modéliser les événements qui n'ont pas été rencontrés lors de la phase d'apprentissage. Pour cela, il utilise une méthode dite de repli (*back-off* en anglais) sur des modèles n-grammes d'ordre inférieur. Ainsi si un mot ne peut être prédit avec un historique de deux mots, alors on utilise la prédiction basée sur un seul mot, c'est-à-dire les bigrammes, et récursivement sur le unigrammes si besoin est. Posons P', la fonction de probabilité calculée en intégrant la formule de Good-Turing et *h'* l'historique réduit du repli. La définition de l'approche de Katz est la suivante :

$$\text{P}(\, \text{m} \mid h \,) = \begin{cases} \text{P}'(\, \text{m} \mid h \,) & \text{si } N(\, h, \text{m} \,) > 0 \\ \alpha(h) \,.\, \lambda(\, h \,) \,.\, \text{P}'(\, \text{m} \mid h' \,) & \end{cases}$$

Équation III.10 : expression de l'approche de Katz

Dans cette équation, $\alpha(h)$ représente un facteur de normalisation assurant que la somme des probabilités du modèle soit égale à 1. Comme cela est indiqué dans [Chen et al. 99], cette méthode est la plus employée de nos jours car elle est efficace pour l'estimation des données manquantes. Il est possible de trouver dans la littérature des approches similaires dont le repli, cependant, se fonde non pas sur des modèles d'ordre inférieur mais sur d'autres types de modèles, comme par exemple les modèles n-classes [Miller et al. 96].

II.3. Autres méthodes

D'autres recherches ont abouti à la définition d'autres méthodes de lissage et de combinaison de modèle pour la prise en compte des événements non appris. D'autres formules de décompte ont été proposées comme pour le lissage linéaire dans [Ney et al. 94]

et de Witten Bell [Witten et al. 91]. Actuellement, le modèle de lissage Kneiser-Ney [Kneser et al. 95] est celui qui donne les meilleurs résultats [Goodman 01]. Pourtant, il n'est pas encore très répandu de par sa relative complexité. Enfin, le lecteur pourra trouver des informations sur le modèle d'interpolation dans [Jelinek et al. 80]. Ce modèle combine tous les niveaux du modèle n-gramme pour le calcul de la probabilité d'une séquence de mots. Ainsi, même si un contexte de longueur *n* n'a pas été rencontré, le score n'est pas nul car il correspond à une somme pondérée des niveaux inférieurs du modèle n-gramme.

Conclusion

Nous avons présenté dans ce chapitre la modélisation statistique du langage ainsi que les modèles les plus courants de nos jours. Le problème crucial des modèles statistiques est celui du manque de données pour l'estimation des probabilités.

Cela étant posé, nous pouvons maintenant affiner nos choix. Nous avons déjà décrit, dans le premier chapitre, pourquoi les modèles de langage statistiques étaient de bons outils dans le cadre de notre étude. De plus, les modèles trigrammes donnent de très bons résultats. D'après la littérature [Rosenfeld 00], les modèles d'ordres supérieurs apportent des résultats encore plus satisfaisants, mais sont nettement plus volumineux [Goodman 01]. Notre choix de modèles trigrammes est aussi motivé par la difficulté de construire automatiquement des modèles n-classes, par manque de corpus étiquetés et de méthodes de classification performantes sur tous types de données, comme nos corpus extraits d'Internet.

En ce qui concerne les techniques de lissage, si l'on se réfère aux conclusions de [Goodman 01], il est évident que, même si des différences de performances sont notées au cours d'expérimentations, aucune ne surclasse vraiment les autres. Cela est renforcé par le fait que les meilleures méthodes sont les plus lourdes et, à l'heure actuelle, sont de ce fait inutilisables dans un système grand public. C'est pourquoi nous utiliserons la méthode la plus populaire depuis des années, l'approche de Katz, qui a l'avantage supplémentaire d'être simple à mettre en œuvre.

Ces choix étant effectués, nous sommes maintenant confronté aux problèmes de corpus pour l'apprentissage de nos modèles de langage statistiques. Nous avons besoin de corpus volumineux pour une bonne estimation de nos probabilités. De plus, il faut avoir une très grande diversité pour augmenter la couverture et la robustesse de nos modèles. Le chapitre suivant présente une étude quantitative et qualitative des corpus que nous extrayons de l'Internet. Nous présenterons leur intérêt dans le cadre du calcul de modèles de langage statistiques.

Partie II : Modélisation automatique du langage à partir d'Internet

Chapitre IV : Corpus tirés d'Internet

Chapitre IV : Corpus tirés d'Internet

Présentation du chapitre

L'ordre de grandeur de la taille des corpus nous permettant d'atteindre nos objectifs peut être estimé en fonction de la taille du vocabulaire. *A priori*, ce besoin en texte est très généralement de plusieurs millions de mots pour des vocabulaires restreints et peu dépasser plusieurs centaines de millions de mots pour des vocabulaires plus importants. Cela est très nettement supérieur à la taille des corpus qu'il est raisonnablement envisageable de collecter en transcrivant des expérimentations de type « magicien d'*Oz* ». Nous avons fait ici un pari essentiel, justifié *a posteriori* par nos résultats : c'est l'hypothèse que "l'écrit peut servir à l'oral", et que du texte libre et spontané, comme on en trouve en très grande quantité sur Internet, peut servir de base à la construction de très bons modèles de langage statistiques.

Ce chapitre présente donc Internet et les données que nous pouvons en extraire pour nos besoins. Il commence par un bref historique et se poursuit par la description des données que nous trouvons actuellement sur le réseau. À ce stade, nous expliquerons quelles sont celles que nous avons décidé d'exploiter et pourquoi. Nous présenterons alors les différents robots de collecte que nous avons développés.

Ensuite, nous passerons en revue les différents corpus que nous avons collectés, et nous les comparerons avec un corpus de texte écrit extrait du quotidien « Le Monde ». Nous mettrons alors en avant les caractéristiques de ces corpus qui nous intéressent principalement pour la modélisation du langage oral.

I. Historique

Internet est l'abréviation de *INTERconnected NETworks* ou de *INTERconnection of NETworks* [Web 04]. C'est actuellement le réseau des réseaux qui relie entre eux des millions d'ordinateurs, donc des millions de personnes, à travers le monde entier. À l'origine, un premier réseau a été développé par les militaires américains pour s'affranchir, en cas de

guerre avec le bloc de l'est, des problèmes liés à une destruction probable des infrastructures de télécommunication standard. L'idée sous-jacente est de construire une structure décentralisée permettant la continuité des communications en cas de destruction partielle, contrairement à l'approche centralisée en place à l'époque. Cette démarche débute à la fin des années 50 par des recherches menées par l'ARPA (Advanced Research Projects Agency) et débouche en 1969 sur la création de l'ARPANET, l'ancêtre de l'Internet. Au cours des années 70, la création du protocole TCP/IP par l'*InterNetWork Working Group* et la décision majeure de l'ARPA de ne pas garder le secret défense sur celui-ci, permirent l'entrée progressive d'un nouveau public, passant des militaires et contractuels de la défense aux chercheurs et universitaires. Au début des années quatre-vingt, la NSF (National Science Fondation) finança le NFSNet et, parallèlement, les militaires créèrent le réseau MILNET. En 1989, le Canada s'équipa à son tour de réseaux universitaires reliés à celui de la NFS. C'est à ce moment que nous pouvons considérer qu'est réellement né ce que nous connaissons aujourd'hui sous le nom d'Internet.

Par la suite, tout s'accéléra. En 1992, le CERN (Centre Européen de Recherche Nucléaire) et plus particulièrement Tim Berners Lee et Robert Caillaud proposèrent le *World Wide Web* comme moyen de diffusion du multimédia sur Internet intégrant textes, images, sons, etc. La notion d'hypertexte fut étendue et devint hypermédia, c'est-à-dire un ensemble de documents multimédias reliés entre eux par une toile, d'où l'utilisation du mot *Web* qui signifie toile en anglais, d'Uniform Resource Locators (URL), des liens définissant universellement chaque document. Dans le même temps, Internet atteint son premier million de machines connectées. En 1993, *Mosaic*, le premier navigateur Web créé par la NCSA (National Center for Supercomputing Applications) fit son apparition. En France, les universités et des centres de recherche comme le CEA et l'INRIA fondèrent le réseau RENATER (Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche). 1994 marque un tournant important pour le développement d'Internet car c'est l'année de l'apparition des fournisseurs d'accès qui permettent enfin à la collectivité de se connecter à Internet. Parallèlement, l'augmentation constante du nombre d'ordinateurs personnels dans les foyers, a fait croître le nombre d'internautes potentiels.

Avec la démocratisation des ordinateurs personnels, l'arrivée du commerce électronique (e-commerce), l'extraordinaire développement du réseau chez les particuliers avec de nouvelles technologies comme l'ADSL, Internet est devenu l'un des principaux médias, plus de 300 millions de machines sont connectées. Les prévisions évoquent un public, ayant accès régulièrement au réseau des réseaux, de près d'un milliard d'internautes en l'an 2005.

De cette diversité de public, allant des universitaires aux particuliers en passant par les entreprises commerciales, émerge une multitude de données sur la Toile, créées dans des buts différents et donc possédant un contenu varié. Cette richesse est celle dont nous allons essayer de tirer parti dans nos travaux de recherche.

II. Observations et prévisions

II.1. Intérêt d'Internet

Nous venons de voir l'histoire de l'Internet et son extraordinaire croissance en termes de nombre de machines et d'internautes connectés. L'utilisation que nous nous proposons de faire d'Internet, dans le cadre de la modélisation statistique du langage, repose sur deux observations principales. La première concerne l'énorme quantité de données présentes sur Internet. Déjà en 1999, le nombre de documents présents sur la Toile était considérable. À cette époque, Altavista, le système de recherche d'informations le plus utilisé sur Internet à ce moment là, indexait plus d'une centaine de millions de pages Web. Les prévisions actuelles s'affolent et l'estimation du nombre de documents disponibles atteint ou dépasse le milliard. Si l'on ajoute à cela leur très grande variété, au niveau du contenu comme au niveau de la forme, il est clair qu'il est intéressant d'étudier les propriétés de tout cela est évident.

Nous pouvons distinguer deux types principaux de documents sur Internet. Les pages dites professionnelles, tout d'abord, présentent des caractéristiques propres au langage écrit de par leur forme et leur contenu. A l'opposé, les pages personnelles sont plus proches d'un dialogue entre l'Internaute qui les a composées et celui qui les consulte. De plus, le tutoiement est généralement de rigueur, la *netiquette* [RFC 1855] précisant d'ailleurs que le tutoiement cordial est préconisé dans les échanges entres les Internautes. Bien sûr, il existe des pages personnelles qui se rapprochent de pages professionnelles. De toute façon, notre préoccupation est de pouvoir trouver toutes les formes de textes en quantité suffisante pour obtenir des modèles de langage efficaces.

Internet ne se résume pourtant pas à la Toile, il existe de nombreux autres services tels que les forums de discussion, les services d'IRC (Interactive Relay Chat) et autres *Chat* en ligne. Nous nous intéresserons donc à tout ce que cette manne de documents textuels peut nous apporter comme information sur la langue française pour l'apprentissage de nos modèles stochastiques.

II.2. Quantification de la part française de la Toile

La quantité, certes intéressante, n'est pourtant pas tout dans la réalisation de nos objectifs. Il nous fallait être certain de trouver suffisamment de documents contenant des textes en français. Dans les travaux du projet Babel [Web 05] de l'Internet Society, nous avons trouvé des statistiques sur la proportion de pages en français sur le Web. Le principe est de déterminer la langue utilisée sur des serveurs pris au hasard sur le réseau. Le parcours des serveurs est fondé sur le tirage aléatoire d'une valeur sur 32 bits, la taille d'une adresse IP. Ensuite, la présence ou l'absence de la machine est vérifiée par une tentative de connexion. Sur toutes les machines ayant répondu, soit 30 millions d'essais pour 60000 machines finalement actives, il reste à trouver celles qui disposent d'un serveur Web. A ce stade, il n'en reste plus que 8000. La méthode consiste à extraire de la page d'accueil de ces serveurs le texte pur en enlevant les balises du langage HTML. Si le texte est suffisamment long (500 caractères au moins), il est alors soumis à un détecteur de langue capable

d'identifier 17 langues.

Bien que, parmi les langues du monde, la langue française ne soit que la 10ème langue parlée, elle arrive au 4ème rang des langues présentes sur le Web, derrière l'allemand et le japonais respectivement 2ème et 3ème, et surtout loin derrière l'anglais, 1er avec 80% des pages. Ces résultats nous ont conforté dans notre première hypothèse qui était de trouver beaucoup de documents français sur le Web. Actuellement, ces statistiques ne sont pas à jour car elles n'ont pas été réactualisées depuis plusieurs années. Cependant, les estimations du moteur de recherche *AllTheWeb* confirment ces résultats.

II.3. Types de données disponibles

Dans cette partie, nous allons indiquer quelles sont les données présentes sur Internet au sens le plus large du terme. Contrairement à une opinion très répandue, il existe de nombreux services autres que ceux permettant l'accès à la Toile et au courrier électronique, qui sont basés sur des protocoles réseaux spécifiques. Dans ce bref aperçu, nous distinguerons deux types de données, statiques et volatiles. Pour diverses raisons qui seront détaillées, nous avons décidé de ne collecter que des données statiques.

II.3.a. Données statiques

Les informations que nous considérons comme statiques sur Internet sont celles auxquelles on peut accéder plusieurs fois dans le temps, et qui possèdent donc une méthode d'adressage intemporelle. Leur contenu peut se modifier au cours du temps mais il reste généralement le même sur plusieurs jours. Nous verrons plus tard que ce point nous est utile pour nos collectes.

II.3.a.1. Documents du World Wide Web

Le World Wide Web, appelé aussi Web ou Toile en français, contient des documents formatés avec des balises HTML. Celles-ci permettent non seulement la présentation de textes avec l'inclusion de documents multimédia, mais aussi des liens vers d'autres documents : les URL. Le passage d'un document à un autre se nomme *navigation*. Les documents ne proposant pas de liens sont des feuilles de la Toile. Cette propriété de navigation est très intéressante car elle permet la découverte, en partant d'un seul document non feuille, de très nombreux autres documents. En partant d'un document situé n'importe où sur la Toile, il est possible d'accéder à des millions d'autres pages Web. Le protocole pour accéder à des documents est un protocole textuel nommé HTTP [RFC 1945], [RFC 2616] pour *HyperText Transfer Protocol*.

II.3.a.1.a. Documents HTML et documents textuels

Ce sont les documents sur lesquels s'est porté immédiatement notre intérêt. En effet, ils correspondent à une grande partie des données qu'il est possible de trouver sur Internet. Ils nécessitent pourtant une analyse et un filtrage particuliers pour enlever le code HTML qui les met en forme. Sur le Web, ces documents représentent de très nombreux domaines d'intérêt et donc de sujets potentiels.

II.3.a.1.b. Documents multimédia

Depuis les début d'Internet, la part des documents multimédia (image, vidéo, son, etc.) n'a cessé de croître. La puissance des ordinateurs familiaux permettant la fabrication personnelle de ces types de documents, la vitesse des réseaux allant croissant et permettant donc de faire voyager ces informations, généralement de très grande taille, de plus en plus vite sont autant de facteurs concourant à ce phénomène. Cependant, ces documents, à de rares exceptions près, ne nous sont d'aucune utilité dans notre tâche de modélisation du langage. Avec l'arrivée de systèmes de reconnaissance performants, il est envisageable de collecter ces documents, de les étiqueter et d'utiliser le texte ainsi obtenu pour nos apprentissages. Cependant, ceux-ci sont vraiment très volumineux et le travail pour réaliser leur étiquetage, puisque nous ne possédons aucune information sur le vocabulaire employé, est titanesque et n'apporterait pas suffisamment de données au regard de l'effort à fournir.

II.3.a.1.c. Newsgroups

Nous utilisons, volontairement et afin d'éviter toute confusion, l'anglicisme *newsgroups* pour désigner les forums de discussion accessibles en mode texte via le protocole NNTP [RFC 977], [RFC 2980] acronyme de *Network News Transfer Protocol*. Il existe maintenant de nombreux forums sur des sites de la Toile, mais certains sont de nature volatile. Les newsgroups ont un fonctionnement différent. Ainsi, un message est envoyé dans un des thèmes prédéfinis sur le serveur. Ensuite, il est propagé de serveur en serveur. Ainsi, chaque serveur possède une copie de tous les messages. La collecte n'en est que plus aisée puisqu'une seule connexion à un seul serveur est nécessaire.

L'intérêt majeur de ces messages est qu'il s'agit très souvent de questions-réponses. Nous trouvons des formes interrogatives et affirmatives en grande quantité. De plus, le tutoiement étant de rigueur, le contenu des newsgroups présente de bonnes caractéristiques pour la modélisation du langage dans un cadre dialogique. Cependant, il faudra vérifier que nous trouvons aussi des formes de vouvoiement pour espérer une bonne couverture dans notre modélisation. Les newsgroups étant aussi hiérarchisés de manière thématique, cette propriété peut se montrer intéressante pour la définition du vocabulaire dédié à une tâche spécifique par exemple.

II.3.a.2. Les autres types de service de distribution de documents statiques

D'autres services utilisent des protocoles différents pour l'accès à des documents de tous types [Tannenbaum 96]. De nos jours, ce sont principalement les serveurs FTP (File Transfer Protocol). Ils nécessitent une gestion de mots de passe pour l'accès aux données. De plus, les données disponibles sont majoritairement des documents multimédia. Enfin, il n'existe pas de méthode de découverte automatique des ces serveurs contrairement au Web.

II.3.b. Données volatiles

Les serveurs IRC et les autres services de discussion en ligne, les CHAT ou *clavardage* et *causette* en français, consistent en des discussions synchrones entre plusieurs participants du monde entier. Ces discussions contiennent vraiment du texte de dialogue en ligne par

l'intermédiaire du clavier. Pourtant, plusieurs problèmes ont mis un frein à leur utilisation. Pour aller plus vite et augmenter l'interactivité, de nombreux raccourcis ont été créés afin d'augmenter la vitesse d' "élocution" des participants, comme par exemple *mdr*, acronyme de « mort de rire » pour indiquer que l'on est en train de rire. La liste des discussions en cours ne permet pas directement d'optimiser la récupération de texte en français. De plus, hormis l'IRC, les autres serveurs de discussion fonctionnent avec des protocoles propriétaires qui ne sont pas accessibles aux développeurs. Enfin, il y a un esprit nettement plus privé à ces discussions en ligne, et cela pose un problème de confidentialité. Contrairement aux pages de la Toile exposées volontairement, les internautes ne s'attendent pas à être "espionnés" sur ces serveurs. Nous avons alors un problème éthique pour l'utilisation de ces données.

II.3.c. Choix effectués

Notre décision devait fournir le compromis maximal entre la taille des données, leur adéquation prévisible à nos besoins, le temps de développement et enfin l'automatisation du processus de collecte. Pour réaliser notre étude, nous avons sélectionné deux sources, sans démentir pour autant la qualité des autres. Nous collectons les documents venant de la Toile et des newsgroups. Effectivement, en limitant la collecte aux domaines Internet francophones pour le Web et aux newsgroups français, il est possible d'optimiser la collecte de données françaises. De plus, les protocoles d'accès sont disponibles car définis par un groupe de travail, le *Network Working Group*. Ces documents étant statiques, la validité d'une référence à ceux-ci est supérieure à quelques jours dans les deux cas, ce qui permet de collecter des données sans problèmes de doublons et de références invalides. Enfin, la taille des données qu'il était prévisible d'obtenir était conséquente.

Ces deux choix nous sont apparus comme fournissant le meilleur compromis entre le travail d'ingénierie qu'il serait nécessaire de fournir et le travail de recherche que l'on pourrait effectuer avec ces données.

III. Robots

III.1. Collectes des documents de la Toile

III.1.a. Intérêt d'un robot propriétaire

La première question qui nous est posée lorsque nous indiquons que nous avons développé notre propre outil de collecte de page Web est "pourquoi avoir réécrit ce genre de programme ?" Plusieurs facteurs techniques sont entrés en jeu. Le principal concernait l'inadéquation des outils existants à nos besoins. En effet, la plupart des collecteurs créent un fichier par page Web, ce qui, pour plusieurs millions de pages, devient un véritable problème de stockage. De plus, à cette époque, les capacités d'analyse des outils existants étaient plus limitées qu'aujourd'hui. Même les plus grands sites d'indexation, comme Altavista, ne suivaient pas les cadres (*Frames* en anglais) ou certaines autres indications de navigation, c'est-à-dire qu'ils ne trouvaient pas tous les liens vers toutes les pages. Nous nous sommes alors décidé à écrire notre propre robot. Nous avons pour cela collaboré avec l'équipe MRIM

(Modélisation et Recherche d'Information Multimédia) de notre laboratoire et plus particulièrement avec Mathias Géry, doctorant en informatique, qui avait le même besoin que nous pour de telles données mais en vue de travaux dans le domaine de la Recherche d'Informations.

III.1.b. Premier robot en PERL

Le premier outil que nous avons développé pour tester la validité de notre démarche était écrit en langage PERL [Wall 00]. Cela nous permettait d'avoir directement de grandes facilités d'analyse, ce langage étant particulièrement bien adapté au traitement de chaînes de caractères à l'aide d'expressions régulières. De plus, il nous fut facile de trouver sur Internet des exemples de programmes réalisant une requête sur un serveur Web pour demander un document. Le principal problème de cet utilitaire était son manque de performance. Il n'était capable que d'une seule demande à la fois et ne pouvait réaliser le traitement des pages déjà collectées pendant les temps de latence entre les arrivées de données en provenance du réseau.

Malgré ses points faibles, cet utilitaire nous a cependant permis de commencer nos recherches et de développer nos premiers outils de traitement des données de la Toile. Il fut aussi, par la facilité d'écriture de règles de traitement, la plateforme de mise au point de l'analyse de nos pages Web.

III.1.c. Robot Clips-Index

III.1.c.1. Description de l'ingénierie mise en oeuvre

Clips-Index [Web 08] est écrit en C++ en environnement Windows. Il est composé d'un grand nombre de processus légers travaillant collaborativement, avec un coordinateur distribuant à chacun le travail à faire. Ainsi, il est possible d'avoir plus de 500 processus légers soit 500 connexions simultanées à 500 serveurs différents. Le point fort de ce collecteur est sa capacité d'analyse des liens au sein des pages Web et sa capacité à suivre les cadres. En effet, de nombreux liens sont mal formés, au sens de la norme HTML, et ne sont pas directement exploitables. Clips-Index inclut un ensemble de règles pour corriger le maximum de ces erreurs et augmenter ainsi ses performances de collecte.

Les capacités de Clips-Index dépendent de la machine sur laquelle il est exécuté. Pourtant, sur un simple ordinateur PC avec un processeur cadencé à 300 MHz et possédant 128 Mo de RAM, le robot est capable de ramener plus de 3 millions de pages par jour. Cette capacité de traitement nous permet d'obtenir des images de l'Internet à un instant donné comprenant plusieurs millions de pages. Par contre, pour une question d'efficacité, la liste complète de ces URLs doit être en mémoire et en accès rapide. Pour ce faire, nous avons dû employer des tableaux en accès semi-direct. Le principe est d'utiliser non pas des nombres pour indexer les positions des éléments dans le tableau, mais des chaînes de caractères. Pour chaque accès au tableau, une clé dite de *hash* ou de hachage est calculée sur la chaîne d'index. Cela permet l'accès direct à une sous partie du tableau. Finalement, pour trouver l'élément recherché, une recherche séquentielle est effectué en comparant les chaînes d'index entre elles.

Les fonctions de hachage que nous avons trouvées sur Internet provoquaient un grand nombre de collisions dans nos accès au tableau. Il fallait donc trouver une fonction limitant ce problème. De plus, dans le fonctionnement standard de ces tableaux, la chaîne d'origine est gardée pour la désambiguïsation séquentielle en cas de doublons de valeur de hash. Cela étant très peu efficace au regard de la consommation mémoire, nous avons opté pour la création d'une clé de hachage sur 64 bits. Pour cela, nous avons compulsé de nombreux algorithmes de hachage et construit le nôtre à partir de plusieurs fonctions de *hash* et ainsi nous avons pu nous passer du stockage de la chaîne concernée. Dans ce cas, la première partie de la clé sur 32 bits permet l'accès direct à une sous-partie du tableau, la seconde servant à différencier les doublons au sein du tableau de hash. L'algorithme de calcul de ces deux clefs est donné, en C++, sur l'exemple IV.1.

```
void CalculCles( char * chaine, unsigned int& cle1, unsigned int& cle2)
{
	// la variable i sert à obtenir un masque de bits pour un & binaire
	// négatif à la première utilisation grâce au --i
	int i = 0;
	// on sauvegarde le caractère précédent
	char precedent = '\0';
	// pour parcourir la chaine
	char * pstr = chaine;
	// les deux sous-clés 32 bits sont initialisées à 0
	cle1 = cle2 = 0;
	do
	{
		cle1 = *pstr ^ (2261933 * cle1 + 446819) % 17176657;
		precedent = *pstr;
		cle2 += *pstr++ << (0xF & --i);
	}
	while (*pstr); // tant que la chaine n'est pas vide
}
```

Exemple IV.1 : code du calcul d'une clé 64 bits utilisée par Clips-Index pour la gestion de ses URL

A l'aide de ce mécanisme de clé codée sur 64 bits, nous possédons une fonction permettant un fonctionnement optimal de notre tableau de hash à la condition que le tableau soit de dimension cohérente avec le nombre d'entrées que l'on désire y ranger. Cette clef nous permet de distinguer sans doublons plus de 5 millions d'URL représentant autant de documents de l'Internet. De plus, le gain en mémoire et en temps de calcul est non négligeable. Cette liste, faisant plus de 250 Mo de texte, est représentée en mémoire par 5 x 10^6 x 4 octets, soit un peu moins de 21 Mo, la clef primaire de 32 bits étant présente dans le mécanisme standard et dans le nôtre. De plus, si l'on obtient, avec la première partie de la clef, une référence déjà existante, il n'est plus nécessaire de réaliser des opérations coûteuses de comparaison de chaînes mais uniquement des comparaisons d'entiers pour réaliser le dédoublonnage éventuel d'entrées dans notre tableau. Ces évolutions ont permit à Clips-Index d'atteindre sa capacité de traitement actuelle.

III.1.c.2. Stratégie de collecte

Au départ, la stratégie de parcours des liens se faisait en largeur d'abord. Ainsi, à chaque tour, le robot ramenait une liste d'URL, traitait tous les documents, en extrayait les nouvelles URLs et traitait l'étape suivante. Un problème apparut lorsque nous avons décidé de respecter un délai entre deux requêtes successives sur un même serveur pour ne pas le surcharger. Dans cette optique, lorsque la liste des URLs se restreint à quelques serveurs, le robot passe le plus clair de son temps à attendre. Pour y remédier, le parcours des liens dans les pages trouvées se fait maintenant dans l'ordre de découverte moyennant les délais d'attente sur les serveurs. Cela permet de ne pas avoir de problème de performance dans nos collectes, sinon tout à la fin, lorsque nous arrivons au point où le robot ne découvre plus aucune nouvelle URL.

Pour finir, comme nous l'avons souligné plus haut dans ce manuscrit, les documents multimédia ne nous intéressent pas directement. Clips-Index possède une stratégie d'élimination de ceux-ci en trois passes. Premièrement, en fonction d'une liste prédéfinie d'extensions (gif, jpg, mpg, etc.), on élimine très simplement ces données multimédia. Mais, avec l'avènement des scripts qui créent à la volée les documents et qui possèdent une extension commune, le problème n'est pas totalement résolu. À l'aide d'une requête d'en-tête pour demander les informations sur le document, Clips-Index interroge le serveur sur le type du document et ne continue son traitement que si ce document est une page écrite en HTML ou un simple texte. Pourtant, ce mécanisme n'est pas toujours suffisant, car certains serveurs, mal configurés, répondent que le document est une page HTML alors que c'est un film par exemple. Pour finir, Clips-Index analyse le contenu de tous les documents pour déterminer rapidement s'il s'agit d'un texte ou non. Nous avons dû définir une mesure suffisamment efficace pour ne pas pénaliser le fonctionnement du robot et permettre l'élimination de ces documents. Cette mesure est basée sur les codes ASCII des octets contenus dans le document. Ainsi, si ce dernier est principalement constitué de caractères imprimables, c'est-à-dire représentant des informations textuelles, nous considérons que c'est un document texte ou HTML puisque le HTML repose sur des balises écrites. Clips-Index compte les caractères dits de contrôle, de code ASCII compris entre 0 et 32, en omettant les caractères 9, 10 et 13 car ils correspondent à la tabulation et au retour à la ligne. Nous avons fixé empiriquement 2% de caractères de contrôle comme seuil au delà duquel nous rejetons le document. Ainsi Clips-Index peut d'une part limiter la bande passante réseau dont il avait besoin en ne la gaspillant pas en téléchargement de documents multimédia qui font souvent plusieurs centaines de Ko, et d'autre part focaliser ses collectes sur des documents nous apportant un maximum d'informations textuelles.

Une dernière mesure a été intégrée au sein de Clips-Index pour limiter la collecte de documents jugés inintéressants. En effet, la recherche de notre robot étant exhaustive, il collectait de nombreux fichier *log*, ne contenant par exemple qu'une liste d'accès au serveur avec des nombres et aucune information critique pour nos besoins. Nous avons fixé un seuil modulable correspondant à la taille maximale des documents que nous ramenons. Dans tous nos corpus, nous avons tronqué toutes les pages Web à 2 Mo au maximum. Ainsi, deux cas peuvent se produire. Soit le document est jugé trop gros mais contient réellement du texte

(texte intégral d'un gros livre par exemple), alors on utilise quand même les 2 premiers Mo de données. Soit, ce n'est pas le cas et nous ne pénalisons pas trop notre espace de stockage et la bande passante réseau en ne ramenant que le début d'un fichier inintéressant. Il est ici utile d'indiquer que la taille moyenne d'un document sur la Toile est, au moment de la rédaction de notre manuscrit, de 8,5 Ko. Ces problèmes restent donc très rares.

III.1.c.3. Conclusion

Clips-Index a été pensé et écrit dans le but bien précis de répondre directement à nos besoins. De plus, il respecte les commandes faites aux robots de ne pas collecter certains documents exprimés dans les fichiers 'robots.txt' [Web 12] présents sur les serveurs Web. Cela nous préserve de l'utilisation de documents, proposés sur la Toile, mais dont l'auteur ne désire pas que le contenu soit exploité. Clips-Index, bien que ses capacités de collecte soient importantes, est un outil non intrusif. Le lecteur trouvera en Annexe G des captures d'écran permettant de voir quels sont les paramètres qu'il est possible de régler et le fonctionnement général de ce logiciel.

III.2. Collecte des messages sur les newsgroups

Contrairement à la Toile, les newsgroups ne fournissent pas immédiatement toute l'information disponible. Les messages sont envoyés au serveur et ont une durée de vie de quelques jours en général. La collecte sur quelques jours, comme pour le Web, n'est donc pas envisageable pour obtenir une grande quantité de données. Dans cette optique, notre outil News-Index réalise une collecte sur une longue durée en se connectant périodiquement sur le serveur afin de connaître la liste des nouveaux messages dans chaque newsgroup. Ce mécanisme exclut toute notion de performance de la collecte, le robot n'interrogeant le serveur que toutes les heures, pour quelques secondes seulement. Cet outil a donc été aussi développé en PERL pour augmenter la rapidité de développement.

Les options proposées concernent uniquement la hiérarchie des newsgroups que l'on souhaite collecter et les filtres avancés à base d'expressions régulières, pour affiner ce choix. Il est ainsi possible de cibler les thèmes auxquels l'on souhaite s'intéresser ainsi que leur langue, lorsque la hiérarchie le propose, ce qui est le cas en français.

IV. Données brutes collectées

IV.1. Données extraites de la Toile

Nous avons réalisé de très nombreuses phases de collecte de documents en provenance du Web. La méthode de choix des domaines Internet à forte composante francophone, ainsi que la liste complète de ces domaines, sont détaillés dans l'annexe A. Nous ne citons ici que les 3 principaux corpus que nous avons au moins partiellement étudiés. Ceux ayant été interrompus pour des raisons techniques, comme un problème de réseau très long, ne sont pas listés. Nous précisons que le temps de collecte de chacun de ces corpus est le même, c'est-à-dire environ 80 heures.

Corpus	Date de collecte	Taille brute	Nombre de documents	Taille moyenne d'un document
WebFr1	février 1999	10 Go	1550000	6,6 Ko
WebFr3	mars 2000	30 Go	3458831	9 Ko
WebFr4	décembre 2000	44 Go	5057642	8,5 Ko

Tableau IV.1 : Informations sur les corpus WebFr1, WebFr3 et WebFr4

La première remarque concerne l'absence de *WebFr2* dans ce tableau. Celui-ci a été collecté puis partiellement analysé, avant qu'un problème technique sur la machine hébergeant les données ne le fasse disparaître. Des travaux, autres que ceux que nous menons sur la modélisation du langage, y faisant référence, nous avons pris la décision de ne pas réutiliser le numéro 2. Pour les 3 corpus restants, nous pouvons noter que la quantité de documents est en augmentation mais aussi que la taille moyenne des documents tend à se stabiliser, de nos jours, vers 8,5 Ko. Nos travaux de recherche se sont principalement portés sur le premier et le dernier, ceux-ci étant les plus à même de mettre en contraste l'évolution de la Toile au cours de la période allant de février 1999 à décembre 2000.

IV.2. Données extraites des newsgroups

Les corpus extraits des newsgroups correspondent à des collectes effectuées sur une durée d'un mois environ. Le tableau ci-après résume les informations concernant nos corpus.

Corpus	Date de collecte	Taille brute	Nombre de documents	Taille moyenne d'un document
NewsFr1	juin 1999	660 Mo	440000	1,5 Ko
NewsFr2	octobre 1999	680 Mo	497583	1,4 Ko

Tableau IV.2 : Informations sur les corpus NewsFr1 et NewsFr2

Actuellement nous ne possédons que deux corpus provenant des newsgroups. Il ont été créés au cours de l'année 1999 et sont tous deux basés sur la hiérarchie '.fr' qui regroupe tous les newsgroups en français. La différence majeure entre ces corpus se trouve principalement dans le nombre de sous-thèmes abordés. *NewsFr1* est composé de 284 thèmes organisés en arborescence de thèmes et de sous-thèmes. *NewsFr2*, lui, en compte 307 ce qui signifie que de nouveaux thèmes de discussion ont fait leur apparition dans la hiérarchie française.

Actuellement, cette hiérarchie comporte 356 thèmes. La liste complète des thèmes de *NewsFr1* et *NewsFr2* se trouve en annexe B.

V. Étude des documents disponibles sur Internet

V.1. Démarche

V.1.a. Phases d'étude

Avant de partir dans de grands développements de logiciels, nous avons préféré utiliser simplement les outils qui étaient à notre disposition pour réaliser une série de tests et ainsi valider nos premières hypothèses sur le contenu de la Toile et des newsgroups. Pour cela, nous avons rapidement ramené un ensemble de quelques milliers de pages Web au moyen de notre premier outil de collecte et d'analyse en scripts PERL. Nous avons analysé le contenu de ces pages, collectées de manière itérative sans stratégie particulière, et mesuré que potentiellement il serait utilisable. Nous avons ensuite, à l'aide de Clips-Index, collecté une plus grande quantité de données et ainsi obtenu plusieurs millions de documents.

V.1.b. Méthode de décomptage des pronoms personnels sujets

Au regard de notre but, la modélisation du langage oral, il fallait que nous puissions mesurer la part des données utiles à cette tâche sur Internet. Pour cela, il a fallu définir une métrique. Nous avons décidé de mesurer le pourcentage d'utilisation de chaque pronom personnel sujet dans nos corpus. On peut juger ainsi de la quantité relative de chaque type de pronoms personnels. Ainsi, nous pouvons comparer ces corpus à d'autres, représentant le français écrit, pour juger de l'intérêt que nous avons à apprendre nos modèles de langage avec des documents provenants d'Internet. Cependant, il reste un problème pour le comptage de ces pronoms. Ainsi, si certains pronoms de posent pas de problème (*je*, *tu*, *il*, *elle*, *on*, *ils*, *elles*), d'autres formes lexicales comme *nous*, *vous* ou *t* (abréviation de *tu*) peuvent revêtir plusieurs fonctions syntaxiques dans une phrase. Le compte de ces formes lexicales pose donc des problèmes d'ambiguïté qu'il nous faut traiter.

Le traitement des différents types de documents pour le compte des pronoms personnels peut être envisagé de plusieurs manières. La première idée que nous avons tenté de mettre en oeuvre est d'utiliser des outils de lemmatisation. Pourtant, de par leur nature, les systèmes à base de règles étaient évidemment inadéquats pour traiter du texte provenant du Web et des newsgroups. Nous nous sommes tourné immédiatement vers des outils probabilistes qui seraient plus enclins à reconnaître ces données. Après quelques recherches, nous avons opté pour le *TreeTagger* de l'université de Stuttgart [Schmid 94] qui proposait une version traitant le français et avait fourni de très bons résultats dans des évaluations sur des textes anglais, ce qui validait la méthode. Pourtant, dès les premiers essais, nous nous sommes très vite rendu compte que les données étaient bien trop bruitées, du fait du très grand nombre de vocables inconnus entre autres, pour que le lemmatiseur puisse réussir à fournir de bons résultats. Dans cette optique, une solution aurait consisté à traiter simplement un sous-ensemble de pages prises au hasard, à les mettre en formes manuellement et à les

lemmatiser ainsi. Avec ce résultat, une simple règle de trois aurait fourni une extrapolation sur le corpus entier. Pourtant, sur un ensemble de données aussi gigantesque que *WebFr*, il est difficile de mettre en forme 1000 pages à la main et d'estimer par la suite que cela est suffisant pour généraliser aux 1,5 millions de pages concernées.

Nous avons alors décidé d'employer une méthode plus simple à mettre en œuvre et qui, de par la quantité de données en entrée du système, réduirait les problèmes liés aux erreurs, celles-ci devenant alors mineures. L'idée fut simplement de compter les formes lexicales représentant des pronoms personnels dans les différents corpus si elles précèdent une autre forme lexicale française connue. Ainsi, dans "je suis ..." *je* est compté une fois, par contre dans "je sing ...", il ne l'est pas car *sing* est l'anglicisme pour chanter, et donc ce contexte ne nous apporte aucune information sur la construction du *je* en français. Pour obtenir une liste de formes lexicales françaises la plus complète possible, nous avons fusionné deux lexiques : BDLex [Perennou et al. 87] et celui de l'Association des Bibliophiles Universels [Web 07]. Le lexique final ainsi obtenu compte plus de 400000 formes différentes. Ce lexique est celui que nous utiliserons dans toutes nos expérimentations.

En étudiant certains résultats pour valider notre méthode, nous nous sommes rendu compte qu'il existait des confusions dans notre algorithme. Si l'on considère le texte commençant par "je t'aime ...", nous trouvons deux formes lexicales pouvant être comptabilisées comme des pronoms personnels sujets : *je* et *t*. Pourtant ce n'est pas le cas de *t*. Par contre, dans "t'es le bienvenu", *t* représente bien le sujet du verbe car c'est une compression de *tu*. La forme lexicale *t* peut poser problème dans nos comptes. Nous avons alors opté pour un développement en spirale sur un sous-ensemble du corpus pour aboutir à l'établissement de l'ensemble des règles suivantes pour compter les pronoms personnels sujets :

- les formes que nous dénombrons sont *je*, *j*, *tu*, *t*, *il*, *elle*, *on*, *nous*, *vous*, *ils*, *elles*. Les formes compressées ne portant pas obligatoirement l'apostrophe, marque de cette compression, nous ne nous fierons qu'à la forme lexicale de base.
- une forme n'est comptée que si elle précède une forme lexicale française connue, longue de plus de deux lettres. Cela évite de compter les *j* et *t* dans les listes de lettres des menus des pages Web par exemple.
- une forme n'est pas comptée si elle est elle-même précédée d'un autre pronom personnel (cas des verbes transitifs par exemple).

L'application de ces règles simples permet de réduire les erreurs mais en aucun cas de les éliminer totalement. Pourtant, ces statistiques permettent d'avoir une idée importante de l'emploi de ces formes sujets avec une forme de texte aussi difficile à analyser que les documents en provenance d'Internet. Le résultat final est donné sous la forme du nombre de ces pronoms personnels sujets en pourcentage du nombre total de mots français, c'est-à-dire dans notre lexique, dans le corpus. Comme nous verrons plus loin, nous pouvons identifier, avec ce comptage de pronoms personnels, deux différentes classes de textes : ceux de type journalistique et ceux contenant en plus une proportion non négligeable de certains pronoms personnels utilisés dans les dialogues.

V.2. Premiers corpus

V.2.a. Descriptions

Pour mettre en contraste nos corpus avec un texte "état de l'art", nous utiliserons un corpus que l'équipe GEOD possède et que nous nommerons *Grace*. Ce corpus est celui donné comme base de test pour l'évaluation des analyseurs de texte au sein de l'action Grace [Web 06] (Grammaires et Ressources pour les Analyseurs de Corpus et leur Évaluation) de l'Aupelf. Ce corpus contient des textes extraits du journal « *Le Monde* », il représente un français plutôt journalistique. Il nous permettra de mettre en évidence certaines spécificités intéressantes de nos corpus pour la modélisation du langage dans le cadre d'un dialogue.

V.2.b. Étude quantitative

Les corpus que nous allons étudier sont *WebFr* et *NewsFr*, que nous appellerons aussi indifféremment *WebFr1* et *NewsFr1* pour les replacer dans l'ordre de collecte de nos corpus. Dans un premier temps, nous nous intéresserons à la notion de quantité brute de données que nous avons dans chacun de nos corpus. Cette information reste imparfaite car les corpus ne nécessitent pas les mêmes traitements avant l'apprentissage des modèles. Elle donne quand même une indication de proportion entre nos différents ensembles de données.

Grace ne fait qu'une quarantaine de Mo mais il est entièrement utilisable pour apprendre des modèles de langage car c'est un texte français. *WebFr* représente une taille totale d'environ 10 Go. Pourtant toutes ces données ne sont pas directement utilisables. Premièrement, il faut enlever les balises du langage HTML pour obtenir le texte. Nous verrons plus loin dans ce manuscrit comment nous exploitons ces balises pour déduire des informations supplémentaires sur la structure du langage contenu dans les pages. De plus, même si le domaine de collecte est le domaine correspondant à une notion géographique française - cela n'étant pas toujours le cas, certaines machines du domaine '.fr' se trouvent physiquement en Angleterre par exemple - ces pages contiennent des données dans diverses langues. Si l'on considère les statistiques du moteur de recherche *AllTheWeb*, nous pouvons compter sur environ 73% de pages françaises. Enfin, le ratio du nombre d'octets de texte par rapport à la taille du document original est simple à calculer. Dans *WebFr*, en moyenne, chaque document fait 6,6 Ko et, après extraction des balises HTML et transcription sous forme de textes des nombres présents, il reste 4,2 Ko de texte. Le ratio de texte dans *WebFr* est d'environ 63%.

Il est aussi possible d'appliquer les mêmes calculs à *NewFr*. Les messages dans les newsgroups sont très souvent courts. De plus, les questions posées sont souvent reprises en début du message de réponse. Nous verrons que cette propriété peut s'avérer intéressante par la suite. Le contenu originel de *NewFr* contient de nombreuses informations représentées dans l'en-tête du message. Le volume utile de texte est donc inférieur en proportion par rapport au corpus précédent. Si l'on extrait des 650 Mo de données pures le texte présent, cela représente 175 Mo. Pourtant, contrairement aux documents de la Toile, ceux-ci sont à presque 100% en français. Le tableau et les histogrammes ci-après récapitulent les calculs et

les valeurs en volume de données d'apprentissage de chacun des corpus. Le pourcentage de texte est le rapport entre la taille moyenne des pages et la taille moyenne du texte extrait par nos filtres. Le pourcentage de texte français, pour *WebFr*, est celui indiqué par le moteur de recherche *AllTheWeb*. Le volume utile correspond à la taille du texte final après suppression de toutes les balises et autres en-têtes.

Corpus	Taille brute	Pourcentage de texte	Données en français	Volume utile final
Grace	40 Mo	100%	100%	40 Mo
WebFr	10 Go	63%	73%	~4,5 Go
NewFr	650 Mo	26%	~100%	175 Mo

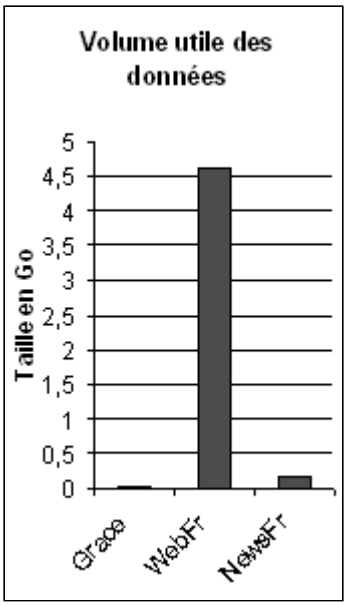


Tableau IV.3 : Figure IV.1 : Volume utile de données françaises dans les corpus Grace, WebFr et NewsFr

Les documents extraits de l'Internet sont nombreux et représentent une masse de données bien supérieure à *Grace* et aux newsgroups, même réunis. Pourtant, même si la quantité de données est importante dans nos travaux, nous avons aussi des besoins en variété et nous ne pouvons exclure un corpus de nos apprentissages en ne considérant que sa taille. Cependant, ces résultats valident notre hypothèse sur la quantité très importante de données qu'il est possible d'extraire de la Toile.

V.2.c. Étude qualitative

L'information qualitative que l'on peut mesurer dans nos corpus doit être de nature linguistique pour être pertinente. Nous avons choisi de mesurer le pourcentage d'utilisation des pronoms personnels français ainsi que le nombre de formes lexicales françaises différentes qu'il est possible de trouver dans chacun de ces corpus. Ces résultats sont ceux que nous avons présentés dans [Vaufreydaz et al. 99c] et [Vaufreydaz 99b].

V.2.c.1. Mesure du pourcentage de pronoms personnels

Les résultats du décompte obtenu pour chacun de nos trois corpus sont présentés, sous formes de pourcentages, sur le graphique IV.2.

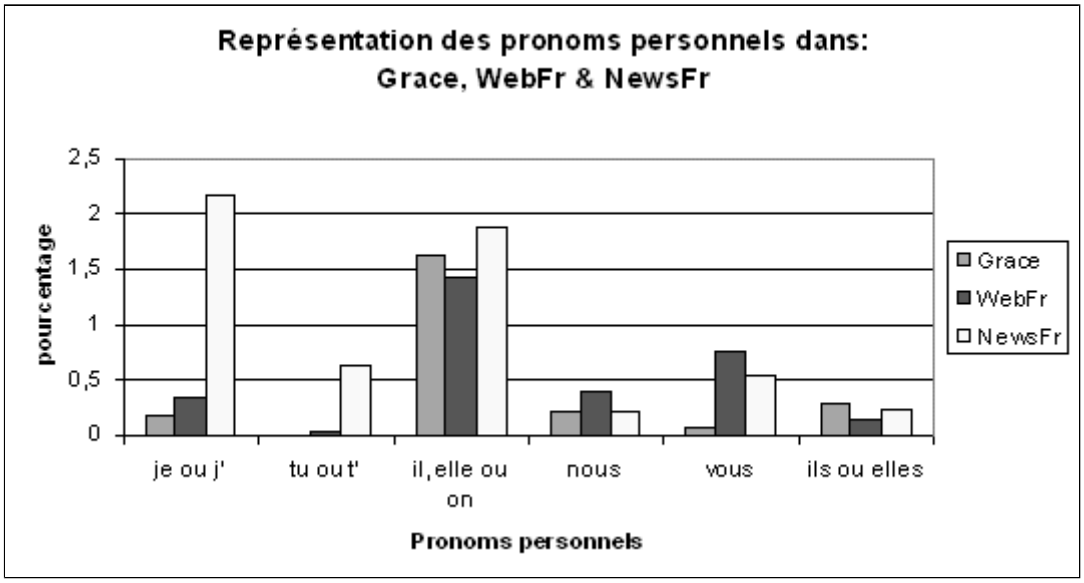


Figure IV.2 : mesure du pourcentage de pronoms personnels dans 3 corpus

Nous pouvons distinguer deux catégories principales de pronoms personnels. Les premiers sont ceux que l'on utilise principalement dans les journaux, c'est-à-dire la 3ème personne du singulier, la 1ère et la 3ème personnes du pluriel. Les autres sont celles que l'on trouve en grande quantité dans les transcriptions des dialogues enregistrés dans le cadre du projet NESPOLE! [Web 02]. Nous nommerons respectivement ces catégories *journalistique* et *dialogue*.

L'analyse catégorie par catégorie du graphique précédent nous apporte beaucoup d'informations concernant nos corpus. Les pronoms de la catégorie journalistique sont bien représentés dans les 3 corpus. Les différences ne sont pas très grandes en termes de pourcentage. Pourtant du fait de la taille de ces corpus, le nombre de représentants dans *WebFr* et *NewsFr* est beaucoup plus important. Nous pouvons donc déduire que, proportionnellement, ces deux corpus sont équivalents, pour l'apprentissage des formes verbales narratives, à *Grace*. Maintenant, si l'on considère la catégorie dialogue, les différences sont bien plus importantes. En ce qui concerne les deux premières personnes du singulier, *NewsFr* en comporte beaucoup plus que *WebFr*, et ce denier plus que *Grace*. Ainsi l'utilisation de *tu* n'existe pas dans *Grace*, ce qui est normal puisque celui-ci est composé d'extraits de journaux.

Si l'on ne s'intéresse qu'à la proportion de chaque catégorie dans nos corpus, nous pouvons classer, en fonction de leur adéquation pour l'apprentissage de formes dialogiques, nos 3 corpus dans l'ordre suivant : *NewsFr*, *WebFr* et *Grace*. Par contre, en termes de quantité de données, l'ordre change car *WebFr* fournit plus de ces formes de par sa taille très importante. De plus, il est évident que l'absence de certaines formes dans *Grace* est problématique. Cela valide notre hypothèse sur le contenu plus dialogique des documents de l'Internet.

V.2.c.2. Nombre de formes lexicales différentes

Nous allons maintenant mesurer la diversité du texte que nous trouvons dans ces corpus. Pour cela, toujours en utilisant notre liste maximale de formes lexicales, nous avons réalisé un simple dénombrement.

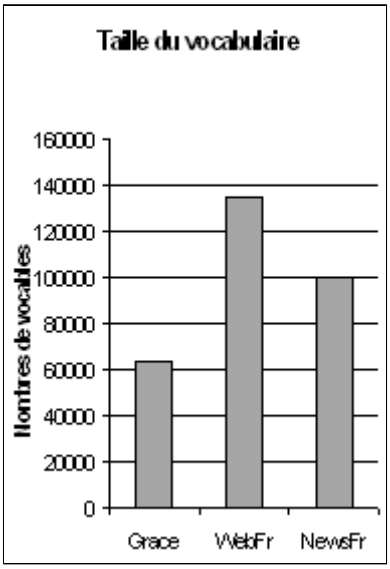


Figure IV.3 : mesure du nombre de vocables différents dans 3 corpus

Comme nous pouvons le voir, les résultats pour chacun des corpus sont très différents. Le moins complet est *Grace*, qui compte près de 63000 formes. Dans *WebFr*, il est possible de trouver 135000 formes lexicales différentes. C'est le corpus le plus varié que nous possédons. Il est à noter que tous les vocables de *Grace* sont inclus dans ceux de *WebFr*, ce qui renforce notre hypothèse de variété de la Toile. En ce qui concerne *NewsFr*, il est constitué d'un peu moins de 100000 mots différents. On retrouve dans les newsgroups du vocabulaire très spécifique du fait des thèmes proposés (philosophie, informatique, histoire, sociologie, médecine, etc.) et cele explique principalement sa diversité par rapport à *Grace*. De plus, il n'est pas totalement inclus dans *WebFr*, il contient plus de 2000 mots supplémentaires. Ceux-ci correspondent aussi à du vocabulaire très spécifique (médical majoritairement), ce qui montre que même si *NewsFr* est beaucoup plus petit que *WebFr*, il apporte une variété qui pourrait être utile si l'on désire réaliser un système de reconnaissance destiné à des médecins.

Ces résultats corroborent notre dernier postulat qui concerne la diversité que peut apporter Internet pour atteindre nos objectifs. Ainsi, comportant de très nombreuses formes lexicales différentes dans beaucoup de thématiques différentes, Internet se révèle être une très grande source de texte pour nos modèles de langage.

VI. Évolution des documents disponibles

Nous avons collecté, au cours de notre doctorat, plusieurs corpus en provenance d'Internet. Nous allons maintenant étudier la progression de la quantité de données

disponibles ainsi que leur variété. Nous ne ferons cette analyse que sur *WebFr1* et *WebFr4*, ceux-ci représentant le premier et le dernier corpus de nos collections. De plus, nous avons mis de côté les newsgroups pour deux raisons. Premièrement, plusieurs sites de la Toile réalisent des interfaces alternatives de consultations des newsgroups et donc ces derniers se retrouvent intégrés dans notre collecte de documents Web. De plus, comme nous le verrons, du fait de la masse de données et de la diversité de *WebFr4*, il est devenu inutile de dissocier ces deux corpus. Ces informations sont présentées dans [Vaufreydaz et al. 01b].

VI.1. Quantité de données

La taille des données collectées est en très nette augmentation par rapport à notre premier corpus. Le nombre de documents disponibles sur la Toile, comme nous l'avions présenti, a été en constante augmentation. Le tableau ci-après rapporte les différents indices de taille de nos deux corpus. Le facteur de 73% de pages françaises est celui indiqué par le moteur de recherche d'informations *AllTheWeb*. Alors, en sachant qu'en moyenne dans *WebFr4*, chaque page fait 8,5 Ko et qu'après filtrage des balises HTML, il reste 43% de texte, nous obtenons les résultats présentés dans le tableau IV.4.

Corpus	**Taille brute**	**Pourcentage de texte**	**Données en français**	**Volume utile final**
WebFr	10 Go	63%	73%	~4,5 Go
WebFr4	44 Go	43%	73%	~13 Go

Tableau IV.4 : Variation de la taille des corpus WebFr1 et WebFr4

Comme nous pouvons le voir, la taille finale de texte en français a été multipliée par 3 environ. Si l'on excepte les approximations, le réel facteur d'accroissement (*fa*) peut être calculé par l'équation
. Notons *MD* la fonction qui donne la taille moyenne d'un document d'un corpus, *MT* celle qui donne du texte extrait, et *T* celle qui renvoie la taille complète du corpus.

$$fa = \frac{\frac{MT(WebFr4)}{MD(WebFr4)}}{\frac{MT(WebFr)}{MD(WebFr)}} \times \frac{T(WebFr4)}{T(WebFr)} = \frac{\frac{3{,}6\ \text{Ko}}{8{,}5\ \text{Ko}}}{\frac{4{,}2\ \text{Ko}}{6{,}6\ \text{Ko}}} \times \frac{44\ \text{Go}}{10\ \text{Go}} = 2{,}92 \approx 3$$

Équation IV.1 : calcul du facteur d'accroissement des données de la Toile

Ce calcul nous donne une bonne idée du premier gain que nous avons en utilisant *WebFr4* plutôt que *WebFr1* pour nos calculs de modèles de langage. Pourtant, il faut noter que la part relative du texte au sein des pages de Toile est en nette diminution, 43% contre 63% précédemment. Les documents que nous avons collectés comportent de plus en plus de données complémentaires qui ne servent qu'à la mise en page et à l'interactivité de la page. Ce sont, en général, des morceaux de programme type javascript qui permettent la

programmation de l'interface, comme des menus déroulant ou des images qui se modifient en fonction de la position de la souris. Il est possible d'expliquer cette évolution par l'apparition de logiciels générant automatiquement ce genre de données au sein des documents en HTML. De plus, les comportements des utilisateurs et leur niveau de compétence ont augmenté ces dernières années. Nous sommes passés de personnes se présentant par un simple texte illustré décrivant leurs passions à des gens proposant des pages très avancées, utilisant des techniques réservées jusqu'alors aux pages professionnelles.

Au final, même si chaque document n'apporte plus que 3,6 Ko de données, le nombre de ces documents ayant considérablement augmenté, le texte total disponible pour nos phases d'apprentissage de modèles de langage a été accru. Nous validons donc notre hypothèse concernant l'intérêt d'utiliser Internet dans le futur, celui-ci proposant de plus en plus de texte.

VI.2. Évolution du pourcentage de pronoms personnels

Nous avons employé la méthode de comptage décrite précédemment dans ce manuscrit. Les résultats sont donnés sous formes d'histogrammes dans le graphique suivant :

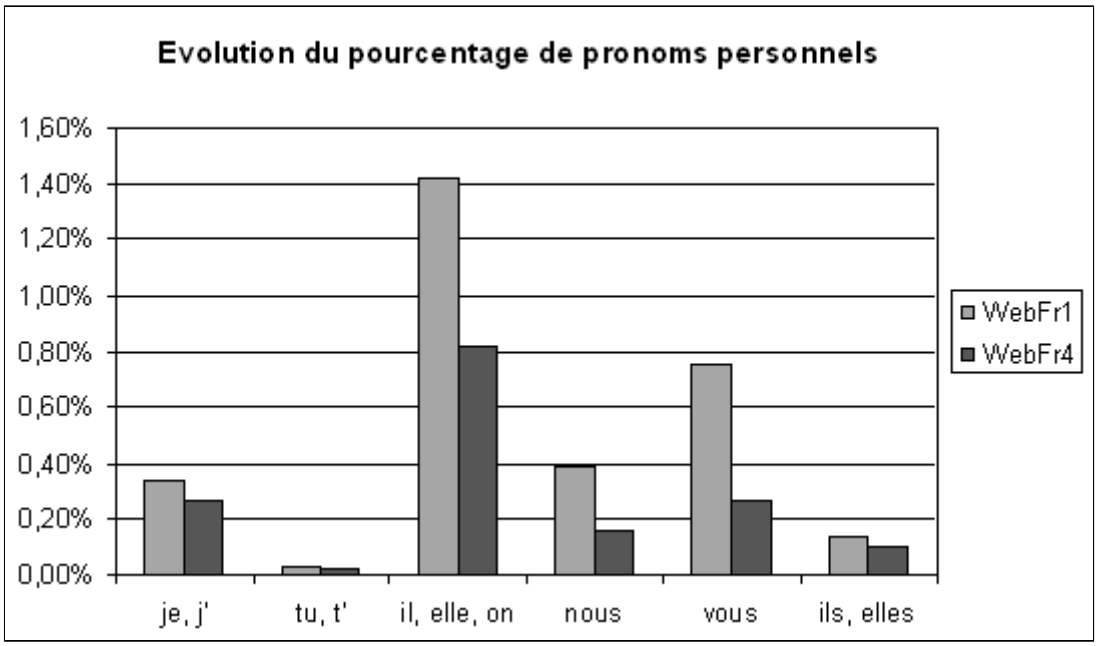


Figure IV.4 : variation du pourcentage des pronoms personnels entre WebFr1 et WebFr4

D'une première observation de ce tableau, nous pouvons affirmer que les différences ne sont pas très grandes pour la 1ère et la 2ème personne du singulier et la 3ème personne du pluriel. Pourtant, il est remarquable que la proportion de ces 3 classes de pronoms personnels a sensiblement diminué. *A contrario*, pour les 3 autres personnes, on note une grande différence dans les pourcentages. La baisse allant jusqu'à plus 50% pour la 1ère personne du pluriel. Nous ne pouvons fournir que des hypothèses sur cette baisse, faute d'avoir le temps et les compétences pour faire une analyse plus profonde des pages que nous proposons. Notre avis est que la qualité du texte présent dans ces documents est meilleure. Ainsi, les phrases

sont plus longues et surtout, les logiciels d'édition de pages Web proposant des outils de correction orthographique, voire de correction grammaticale, la part des erreurs de frappe ou d'étourderie a diminué, alors qu'elles étaient légion dans *WebFr1*. Il en va de même pour le nombre de mots non reconnus comme étant des mots français. Cela modifie nettement les résultats de nos calculs puisque nous ne comptions que le rapport entre le nombre de ces pronoms et le nombre total de vocables en français. Pourtant, nous ne pouvons pas garantir que cette explication soit suffisante pour expliquer cette baisse. Pour l'instant, nous ne pouvons que la constater et nous verrons plus tard dans ce manuscrit, l'influence de ce changement sur les performances de nos modèles de langage dans notre système de reconnaissance.

VI.3. Évolution du nombre de formes lexicales

L'évolution de l'Internet en termes de quantité s'est aussi accompagnée d'une modification du contenu, comme nous avons commencé à le voir avec le pourcentage de pronoms personnels. Nous avons réalisé un nouveau calcul pour connaître les formes lexicales présentes dans nos corpus.

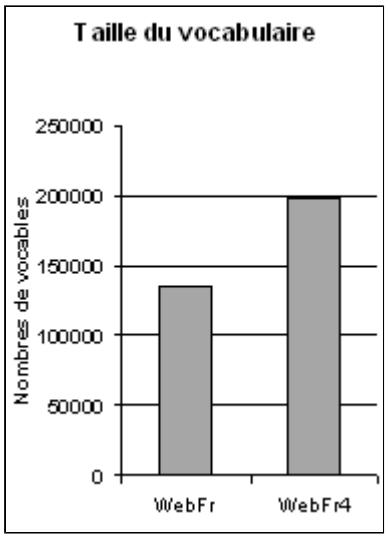


Figure IV.5 : variation du nombre de vocables différents entre WebFr1 et WebFr4

Le graphique montre bien une évolution du vocabulaire, celui-ci passant de 135000 formes environ dans *WebFr1* à presque 200000 formes dans *WebFr4*. L'inclusion des newsgroups dans nos collectes de la Toile, par le biais des sites les archivant, ne suffit pas à expliquer à elle seule cette augmentation. Nous pouvons penser que, le nombre d'internautes grandissant, les thèmes exposés dans les documents sur Internet se sont considérablement diversifiés. Le vocabulaire associé a suivi la même évolution. De plus, comme nous le soulignions déjà à propos des pronoms personnels, les outils d'édition de documents HTML corrigent les fautes commises et donc augmentent la probabilité d'obtenir de la diversité.

Conclusion

Le début de nos travaux était fondé sur deux postulats principaux. Premièrement, le volume et la diversité des documents disponibles sur l'Internet devaient nous fournir des données intéressantes pour nos modèles de langage. Et, comme nous l'avons vu, tant au niveau de l'utilisation des pronoms personnels sujets, surtout pour les formes employées dans les dialogues, qu'à celui du nombre de formes lexicales différentes, *WebFr1* et *NewsFr1* semblent être de très bonnes sources de texte pour réaliser nos apprentissages. Ce premier postulat est vérifié par nos statistiques. Le second prévoyait l'extension du nombre des documents sur Internet et surtout de leur diversité. La comparaison entre *WebFr1* et *WebFr4* montre bien que non seulement nous avons beaucoup plus de données disponibles mais que le nombre de vocables différents a aussi augmenté. Cette seconde hypothèse est donc aussi validée.

Les statistiques corroborent nos prévisions. Pourtant, cela ne nous donne aucune garantie d'obtenir de bons modèles de langage directement avec ces données. Comme nous l'avons signalé, elles sont très bruitées et contiennent, en plus des balises HTML et du code javascript par exemple, du texte dans d'autres langues que le français. Cependant, les résultats que nous venons de présenter fournissent une bonne indication sur l'adéquation de cette forme particulière d'écrit, que l'on trouve sur Internet, pour apprendre des modèles utilisables pour la reconnaissance du langage oral.

Le traitement, la mise en adéquation de ces données et le calcul de modèles de langage font l'objet du chapitre suivant.

Chapitre V : Construction automatique de modèles de langage

Chapitre V : Construction automatique de modèles de langage

Présentation du chapitre

Ce chapitre présente notre méthode de construction automatique de modèles de langage en utilisant des documents en provenance d'Internet. Nous décrirons comment ils sont filtrés pour obtenir du texte, puis comment ce texte, ou tout autre texte, peut être à son tour filtré pour obtenir ce que nous nommerons des blocs minimaux. Nous aborderons aussi le problème de l'extension du vocabulaire et comment nous l'avons traité. Enfin, nous expliquerons pourquoi les outils standard de calcul de modèles de langage ne sont pas adéquats pour l'apprentissage sur un corpus de blocs minimaux. Nous présenterons donc l'adaptation des calculs que nous avons dû faire.

I. Définition d'une tâche de modélisation du langage

Avant de décrire notre procédure pour construire nos modèles de langage, il est nécessaire que nous définissions ce que nous appelons tâche de modélisation. Pour un être humain, une tâche comme la réservation touristique est un ensemble bien défini de situations, de mots, et surtout de contextes où employer ces mots. Pourtant, nous l'avons déjà vu, le modèle de langage statistique est un outil très simple ne permettant que de connaître des probabilités de cooccurrence de mots. On peut aussi ajouter à cela qu'il est très difficile de dénombrer automatiquement, et même manuellement dans certains cas, les contextes d'emploi des mots d'une tâche qui sont utiles pour cette tâche. Dans le cas d'une tâche comme le sport, il existe de nombreuses métaphores très imagées qui empruntent des termes qui n'ont, au départ, aucun lien avec le sport. Dans ce cas, il faut avoir une très bonne

connaissance de la tâche pour pouvoir choisir de garder tel ou tel contexte au sein du modèle de langage. Nous voulions une approche la plus automatique possible en évitant le recours à des experts linguistes. Il s'avère impossible de trouver des experts d'un domaine et de leur faire classer des documents manuellement, par exemple, pour obtenir dans des délais raisonnables une définition précise de la tâche.

La définition d'une tâche, telle que nous l'utiliserons dans le reste de ce manuscrit, est une version moins contrainte que celle d'un humain. Pour nous, une tâche sera un ensemble de termes et tous les contextes dans lesquels nous pourrons les trouver dans les documents en provenance d'Internet. Ce choix s'explique par la diversité des documents de l'Internet que nous avons démontrée dans le chapitre précédent. En ne filtrant pas les contextes rencontrés, nous posons comme postulat que nous aurons un maximum de contextes liés à la tâche et des contextes ne nous servant pas. Ces derniers ne seront de toute façon pas employés par les utilisateurs du système de reconnaissance dans le cadre de la tâche considérée, ils ne constituent alors qu'une perte de mémoire. De plus, si pour une raison quelconque, comme le stress, ce même utilisateur emploie d'autres constructions de mots non corrélées avec la tâche, alors ces contextes supplémentaires augmentent la robustesse du système, et deviennent par là même, très importants.

Bien entendu, nous ne pouvons en aucun cas garantir que nous trouverons tous les contextes de tous les mots de notre tâche dans les documents d'Internet. La capacité de couverture des textes trouvés sur Internet fera partie d'expérimentations dans le chapitre suivant.

II. Description schématique

Nous allons maintenant présenter le cheminement complet qui nous permet de construire à partir d'un ensemble de documents bruités c'est-à-dire ne contenant pas que du texte, des modèles de langage n-grammes. Le synoptique de la figure V.1 présente l'ensemble de la démarche. Les boîtes dont le contour est en trait plein représentent des données que nous possédons (corpus, vocabulaire, etc.), celles en pointillés les données, temporaires ou non, générées par nos outils. Les ellipses correspondent aux différentes phases de la procédure, qui seront détaillées une par une dans la suite de ce chapitre.

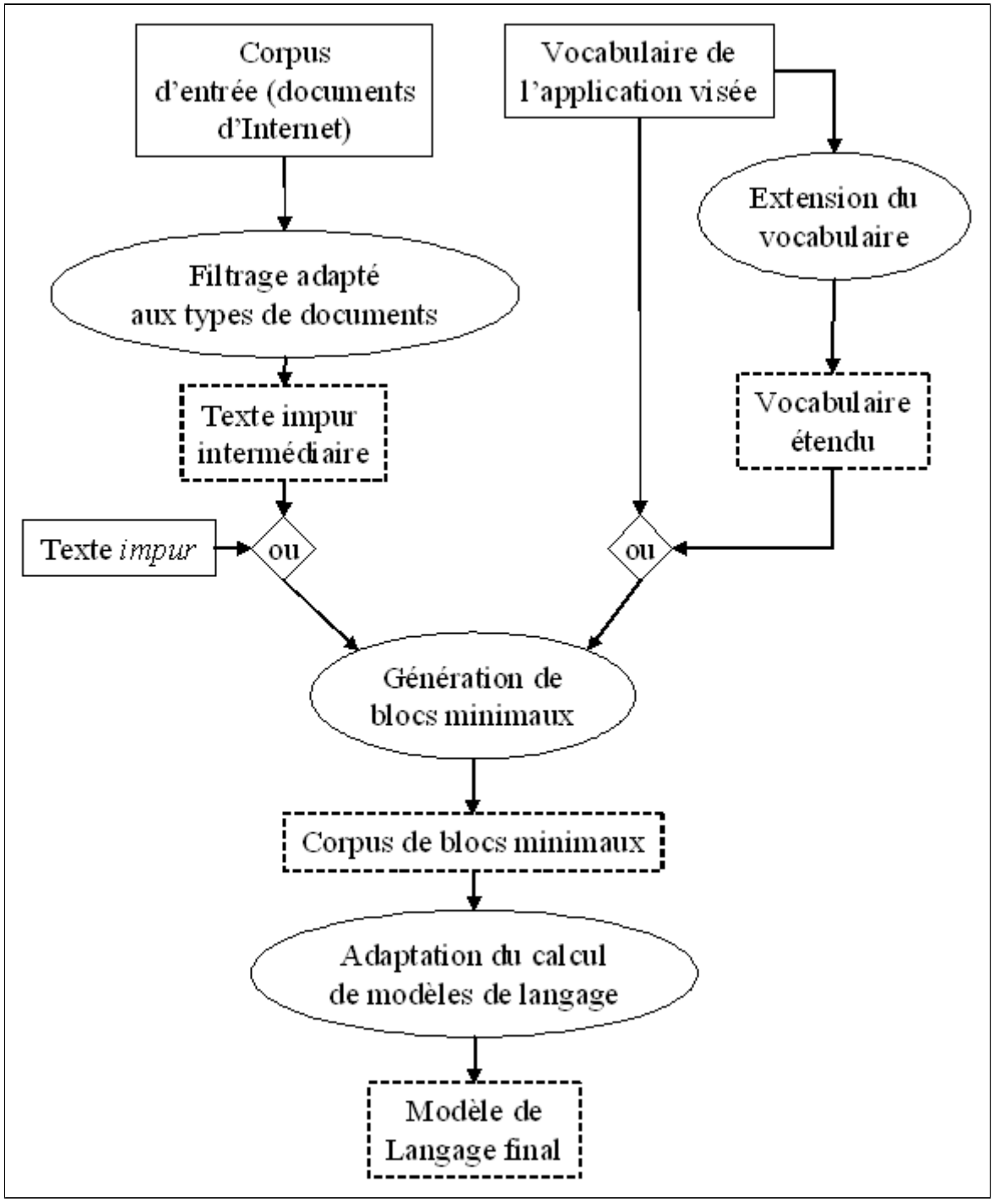


Figure V.1 : synoptique de la construction de corpus

Notre procédé de génération d'un modèle de langage peut se résumer au schéma précédent. Dans un premier temps, il faut filtrer les documents en provenance d'Internet. On obtient alors un texte *impur*. Un texte impur correspond à un texte dont le vocabulaire est supérieur à celui dont nous avons besoin dans l'application. C'est évidement le cas des documents de la Toile par exemple. Si l'on prend *WebFr4*, il est possible d'y compter 6534870 vocables, cependant seulement un peu moins de 200000 sont des formes fléchies françaises. De plus, tous ces vocables ne sont pas utiles dans notre système. Dans le cas du projet CSTAR, le vocabulaire employé est de l'ordre de 2500 mots. Pour le projet Nespole!

cette valeur atteint un peu plus de 20000. Dans les documents d'Internet, une grande masse de données n'est alors pas intéressante pour nous.

Le second point est l'extension du vocabulaire de la tâche. Le principal problème des systèmes de reconnaissance de la parole concerne les mots hors vocabulaire. Ainsi, si un utilisateur prononce un mot que le système ne connaît pas, il est alors remplacé par un ou plusieurs mots connus qui s'en rapprochent acoustiquement. Le problème est que l'insertion de mots approchants peut engendrer des problèmes si les contextes de ce qui a été produit ne sont pas compatibles avec le mot prononcé à l'origine, si par exemple un adverbe est inséré à la place d'une forme verbale. Le système cumule alors plusieurs erreurs à la suite. Pour pallier cela, il est possible d'inclure des mots courants pour minimiser le nombre de mots hors vocabulaire que sont susceptibles de prononcer les utilisateurs. Nous verrons aussi en détail, plus loin dans ce manuscrit, qu'il peut être utile de rajouter des mots composés comme un seul élément du vocabulaire pour augmenter la robustesse.

À ce stade, nous possédons un texte impur plus le vocabulaire final de l'application. La création du corpus pour l'apprentissage passe donc par un filtre qui génère ce que nous appelons des blocs minimaux. Pour résumer, cette technique correspond à une fenêtre glissante d'analyse qui découpe le texte impur en une suite de blocs considérés comme minimaux. Cette appellation tient seulement au paramètre principal du filtre : la longueur minimale de chacun de ces blocs. Ces blocs sont constitués uniquement de séquences de mots qui sont dans le vocabulaire.

Avec ce corpus de blocs minimaux, nous pouvons maintenant finaliser notre travail par le calcul d'un modèle de langage. Cependant, de par la forme particulière du corpus d'apprentissage, nous avons dû réaliser des adaptations dans le calcul de ces modèles. En effet, contrairement à un texte monobloc, les césures entres les blocs introduisent des changements dans la répartition de la masse de probabilité qu'il est nécessaire de prendre en compte.

Nous allons maintenant détailler chaque procédure séparément.

III. Filtrage adapté aux types de documents

III.1. Documents de la Toile

Le filtrage de documents du *World Wide Web* nécessite une bonne connaissance de la norme HTML [RFC 1866] et de la signification de chaque balise présente dans les pages Web. En Annexe C, le lecteur trouvera un exemple de page Web, sa source HTML complète ainsi que le texte qu'il est possible d'en extraire.

Le problème ne se résume pas à une simple élimination des balises HTML. Si l'on ne fait que cela, de nombreux problèmes surviennent. Des mots se retrouvent concaténés car la balise qui définissait leur mode de présentation est simplement supprimée. De plus, la structure des pages peut nous apporter de l'information supplémentaire. Il est possible de déduire de certains éléments de la structure de la page Web, comme le titre, des fins de phrases implicites qui peuvent nous être utiles dans notre tâche de modélisation du langage.

Dans ce cas, tout titre est considéré comme se terminant par un point.

Nous avons opté pour un développement en cycle pour concevoir les règles qui allaient nous servir à filtrer ce type de texte. Pour mettre au point ce filtre, nous avons commencé par le développement d'un programme en langage de script PERL. Celui-ci nous a permis de choisir l'ensemble de nos règles. Celles-ci sont données, par le biais du fichier de configuration de notre outil final, en Annexe D. Il a 3 règles de base :

- la première concerne les balises qui sont simplement supprimées. Ce sont majoritairement les balises entrant en jeu dans la typographie du document qui engendrerait, si elle était remplacée par un espace, des segmentations de mots. Par exemple, dans le cas où l'on aurait « bonjour », décrit dans la page HTML par "`<U>b</U>onjour`", le résultat du filtre est alors "bonjour".
- la seconde est exactement l'inverse de la précédente. Certaines balises indiquent implicitement un espace entre deux mots. C'est le cas pour celles qui délimitent des blocs de texte ou qui intègrent des objets. Dans le cas du début d'une liste à puces comme "`au revoir<LI>bonjour`", le résultat de l'extraction est "au revoir bonjour".
- la dernière extrapole des marques de fin de phrase à partir de certaines autres balises. Dans le cas d'un titre, le code HTML "`bonjour</TITRE>`" est transformé en "bonjour .".

Le document lui-même est considéré comme une entité complète et donc, systématiquement, un point est inséré à la fin du traitement de chaque page. Cela peut provoquer des doublons de marques de fin de phrase mais ceux-ci seront filtrés par le programme de production du corpus final. Notre filtre réalise, en plus, une séparation de tous les signes de ponctuation. Ceux-ci sont donc toujours entourés d'espaces. La dernière modification de notre filtre concerne la réécriture des nombres en toutes lettres. En français, le nombre de mots pour exprimer des nombres est limité. Nous transcrivons tous les nombres rencontrés sur Internet dans leur forme textuelle équivalente. Mais de très nombreux chiffres présents sont formatés et groupés par milliers. Donc un nombre comme 1 million est très souvent écrit comme "1 000 000". Notre programme traite ce genre de nombre en concaténant les chiffres pour obtenir la version correcte des mots. Il intègre aussi certains formats de date et de nombre pour les transcrire au plus juste : "01:30" devient "une heure trente minutes". De plus, des nombres comme 1800 peuvent être transcrits de deux manières différentes : soit "mille huit cents" soit "dix-huit cents". Le filtre génère à tour de rôle chacune de ces formes pour espérer obtenir 50% de chacune d'elles.

Une fois l'ensemble des règles défini, nous avons écrit, pour accélérer les traitements, un programme en C les appliquant : *Html2Text*. En effet, l'utilisation d'expressions régulières dans le langage PERL engendre un très grand nombre d'allocations et de retassement de mémoire. *Html2Text* fonctionne avec un espace mémoire constant et ne réalise qu'un seul parcours de celle-ci à chaque nouvelle page. Ainsi le temps de traitement de *WebFr*, sur le même ordinateur, est passé d'une semaine à quelques heures.

À la fin, nous obtenons un texte continu où les fins de ligne ne sont là que pour faciliter la lecture, dans un logiciel standard d'édition de fichier textuel. Le résultat de ce filtrage, sur le début de la page d'accueil du laboratoire CLIPS, est présenté comme exemple ci-après.

clips - communication langagière et interaction personne - système . actualité . recital moins deux mille cstar présentation . multicom . équipes arcade geod geta iihm mrim trilan . personnes . infos pratiques plan d' accès . créé en dix-neuf cent quatre-vingt-quinze le laboratoire clips communication langagière et interaction personne - système s' est constitué autour de thèmes concernant les interfaces homme - machine les système intéractifs les systèmes multimédia les réalités virtuelles . ses axes de recherche sont . langue traduction et dialogue la langue comme objet d' étude mais aussi comme mode de communication dans le dialogue homme - machine ou la traduction automatique équipes geta trilan geod . systèmes d' interaction les systèmes d' interaction interfaces multimodales réalités virtuelles télé - présence etc . pour des usages finalisés création artistique conception assistée etc . équipes iihm . systèmes multimédias les systèmes multimédias systèmes à base de connaissance systèmes d' information utilisant la langue naturelle environnements hypermédia etc . fournissant les modèles et les outils de base équipes mrim arcade . les thèmes de recherche choisis nécessitent une articulation entre la recherche amont et la prise en compte des usages et des besoins conception valuation pré - développement etc .

Exemple V.1 : extraction du texte contenu dans la page d'accueil du laboratoire Clips

Nous pouvons faire quelques remarques sur ce texte. Tous les mots, hormis les nombres composés générés par le filtre, ont été séparés. C'est le cas de "pré - développement", et de "etc .". Ceux-ci, dans le cas de leur existence dans le vocabulaire, comme les acronymes tels que "s.n.c.f" seront concaténés lors de la phase de calcul de l'extraction des blocs minimaux avec tous les autres mots composés.

III.2. Documents des newsgroups

Ce type de document est beaucoup plus facile à analyser que les pages Web. En effet, ils sont constitués d'une en-tête bien définie avec des champs connus et d'un corps au format texte. Les autres formats de contenu comme le HTML sont proscrits et ne doivent pas être employés même si les outils de consultation des *newsgroups* l'autorisent, comme le précise la *netiquette* [RFC 1855].

La seule difficulté présente dans ces documents reste l'inclusion d'un message dans un autre. Dans les *newsgroups*, lorsque l'on répond à un message, celui-ci est inclus dans la réponse en totalité ou en partie. Les portions du message incluses commencent toutes par le caractère '>'. D'autres caractères peuvent le remplacer mais cela est encore une fois déconseillé par la netiquette. L'extraction du texte d'un message est alors très simple. Au commencement, on identifie le champ titre dans l'entête du message. On considère que le titre est obligatoirement terminé par une marque de fin de phrase. On poursuit en enlevant les marques de citation d'un message précédent s'il y a lieu. Il ne reste ensuite qu'à réaliser la transcription des nombres et la séparation des marques de ponctuation. Nous obtenons un texte sous la même forme que précédemment. Un exemple complet est fourni en annexe E.

III.3. Autres documents

Tous les types de documents peuvent être employés dans notre méthode de calcul, tant qu'ils sont mis sous la forme prévue. Ceci laisse donc ouvert notre système pour l'intégration de tout texte dit *impur* pour le calcul de modèles de langage statistiques.

IV. Extension du vocabulaire

IV.1. Mots hors vocabulaire

Le problème majeur dans les systèmes de reconnaissance du langage est le vocabulaire clos. En effet, que ce soit acoustiquement ou dans le module de modélisation du langage, le nombre de mots est fixe. Dans ce cas, si l'utilisateur prononce un mot qui n'existe pas au sein du système, comme des noms propres ou des noms communs non inclus dans la tâche, il est alors substitué à une séquence de mots proche acoustiquement et probable selon le modèle de langage. Cela peut conduire à une ou plusieurs erreurs consécutives et influencer la reconnaissance des mots suivants.

Diverses approches ont été proposées ces dernières années. Par exemple, des modèles hiérarchiques génériques sont utilisés pour la gestion des noms propres en japonais [Tanigaki et al. 00]. Pourtant cela n'est pas réalisable en français car les noms propres ne présentent pas une structure figée. Cela implique que sans règles de construction, l'espace de recherche représenté par les mots hors vocabulaire est infini.

Ne pouvant modéliser une liste infinie de noms propres, nous avons décidé de ne travailler que sur des formes lexicales du français qu'il nous est possible d'inclure aisément. Avec le décompte de formes lexicales que nous avons effectué sur *WebFr4*, il est facile d'augmenter le vocabulaire avec les formes les plus utilisées sur la Toile. Le premier problème lié à l'accroissement de la taille du vocabulaire est l'augmentation de la taille du modèle de langage. Ainsi en passant arbitrairement de 2500 mots à 20000 par cette technique, le modèle passe de 190 à 242 Mo. L'espace de recherche des algorithmes croit proportionnellement, ce qui allonge le temps de reconnaissance. Le second problème concerne l'apprentissage du modèle statistique. Le besoin en corpus pour l'estimation des probabilités pour un vocabulaire de 20000 mots est nettement supérieur. Cependant, en utilisant les mots les plus fréquemment employés dans le corpus *WebFr4*, nous maximisons l'espérance d'extraire beaucoup de données de ce corpus avec ce vocabulaire.

Il est aussi possible d'augmenter le vocabulaire avec les synonymes des mots présents dans le vocabulaire de base en utilisant, par exemple, des outils comme les espaces sémantiques décrits par [Ploux et al. 98]. Cependant, la notion de synonymie est très dépendante de la tâche et nécessite l'avis d'un expert. Notre méthode se voulant entièrement automatique, nous n'avons pas utilisé cette possibilité.

En conclusion, nous pouvons affirmer que cette technique permet de limiter l'utilisation de formes lexicales non connues par le système, tout en garantissant l'obtention de modèles de langage appris sur une masse importante de données. Le nombre de mots est

choisi empiriquement selon la couverture désirée et selon le système de reconnaissance visé. Dans le cas d'un système embarqué, où la mémoire et la puissance de calcul ne sont pas extensibles à volonté, l'accroissement du vocabulaire peut se limiter à quelques centaines de mots.

IV.2. Mots composés

Dans un système de reconnaissance de parole spontanée pour le dialogue les lettres isolées sont parties intégrante du vocabulaire. Cela permet par exemple au locuteur d'épeler des noms. Le problème se pose lors de la phase de reconnaissance. Le système se retrouve avec des phonèmes et suites de lettres souvent probables dans le modèle de langage. Il génère alors un enchaînement de lettres représentant phonétiquement le signal et ne produit pas de mots longs. Pour pallier cela, de nombreux systèmes de reconnaissance, dont le nôtre, intègrent une pénalité sur l'insertion de mot. Simplement, lorsqu'un nouveau mot est inséré dans une hypothèse, le score de celle-ci est modifié par le modèle de langage et cette pénalité. Ainsi, il est possible de réguler l'apparition des mots en choisissant, souvent empiriquement ou expérimentalement, cette pénalité. Pour un même signal, l'utilisation de cette pénalité, favorise alors les séquences de mots plus courtes dans le cas où leurs scores, selon le modèle de langage, seraient équivalents.

Cette méthode fonctionne parfaitement. Cependant, il existe un revers. Si l'on prend la locution "il y a", dans certaines conditions de reconnaissance acoustique et selon les mots présents dans le vocabulaire, le score cumulé de la partie acoustique, du modèle de langage et de la pénalité d'insertion de mot peut être inférieur à celui d'une autre séquence de mot comme "il a". Dans ce cas, il est difficile d'obtenir du système de reconnaissance cette locution. Pour cela, l'idée de nombreuses équipes de recherche est d'intégrer, au sein du système, la locution comme un seul mot : c'est que l'on nomme les mots composés, *compound words*, *multi words* ou *phrases* en anglais. Un autre avantage d'utiliser des mots composés est de pouvoir plus finement gérer certains phénomènes de co-articulation acoustique entre les mots composant le groupe, simplement en réalisant les variantes phonétiques adéquates dans le dictionnaire acoustique du système de reconnaissance.

Dans la littérature, nous trouvons diverses approches pour le choix de ces mots composés. La majorité d'entre elles sont basées sur des approches statistiques comme la fréquence des bigrammes ou l'information mutuelle des mots [Beaujard et al. 99]. Pourtant, comme le souligne [Saon et al. 99], certains mots très probables, s'ils sont intégrés dans des mots composés, augmentent l'incertitude entre ce mot et les mots composés qui le contiennent. Ce phénomène est encore accentué si ce mot est court, c'est le cas pour 'a' ou 'à' en français par exemple.

En tenant compte de la nature des données que nous possédons et de leur taille, nous avons décidé de baser notre construction de mots composés sur une approche purement statistique. Cependant, nous ne nous contenterons pas de bigrammes mais nous chercherons des mots composés d'ordres supérieurs. De plus, comme nous l'avons dit au début de cette section, le problème est majoritairement posé pour des mots courts, au sens acoustique du terme. Alors, nous limiterons la construction de mots composés aux seuls mots courts. Nous

utilisons les outils standard [Clarkson et al. 97] de construction de modèles n-grammes en choisissant le mode pentagramme. Par la suite, pour chaque rang du modèle, des bigrammes aux pentagrammes, on sélectionne les plus probables et on obtient la liste finale des mots composés. Bien entendu, il est possible par cette méthode de composer des mots plus longs. Cela permet alors, dans un trigramme basé sur de telles compositions, d'augmenter la taille réelle de l'historique considéré. Si "il_y_a" est intégré au vocabulaire, le trigramme "il_y_a une maison" est en fait un pentagramme. Cela augmente alors la robustesse du modèle de langage.

En poussant l'utilisation de mots composés à son extrême, c'est-à-dire en composant tous les mots, il est possible de construire des modèles comme les n-SeqGrammes [Zitouni 00], c'est-à-dire un modèle n-gramme dont le vocabulaire contient une liste maximale de séquence comme entrée du vocabulaire. Son approche est de combiner les classes syntaxiques des mots selon leur information mutuelle. Ensuite, les séquences de mots correspondant à ces classes sont extraites et rajoutées au vocabulaire. Un nouveau modèle de langage est calculé et sa perplexité est mesurée sur un corpus de test. Si elle diminue, l'algorithme itère pour ajouter encore de nouvelles séquences. L'auteur de ce modèle construit des modèles dit n-SeqClasses sur le même principe pour construire des modèles à base de séquences de classes.

Pour conclure, nous ne fournissons pas dans ce manuscrit de liste complète de mots composés que l'on pourrait trouver avec cette méthode et cela pour plusieurs raisons. La première est que nous n'avons calculé des modèles de langage qu'avec des vocabulaires dédiés à des applications spécifiques que nous visions. Le résultat ne serait donc pas directement utilisable par les lecteurs. Ensuite, nous ne l'avons fait que pour des mots courts, 3 ou 4 phonèmes maximum. Enfin, si l'on cherchait à calculer une liste maximale des mots composés dans les documents d'Internet en travaillant avec notre liste de 200000 formes fléchies française, le problème serait ensuite de sélectionner correctement, par seuillage ou manuellement, uniquement ceux qui sont vraiment intéressants ou pertinents. Le lecteur trouvera cependant des exemples de mots composés en Annexe F.

V. Génération de blocs minimaux

V.1. Définition d'un bloc minimal

Nous possédons maintenant les deux parties nécessaires à la génération du corpus d'apprentissage de modèle de langage : du texte impur et le vocabulaire de l'application. Nous allons devoir définir une méthode d'extraction de texte adapté au calcul de modèles de langage n-grammes. L'idée majeure est de travailler sur une séquence de graphèmes contenant bien d'autres formes que celles dont nous avons besoin et d'en extraire une substance adéquate.

Nous avons défini pour cela la notion d'extraction de blocs minimaux. Un bloc minimal est une sous-séquence de la séquence d'entrée du filtre, ne contenant que des mots de l'application visée. Il n'est considéré minimal que s'il contient au moins *n* mots

consécutifs, *n* étant réglable à volonté. De plus, toute fin de phrase, explicite ou implicite génère une fin de bloc, et l'amorce d'un nouveau bloc, virtuel tant qu'il n'a pas atteint la longueur requise. Lorsqu'un bloc remplit les critères demandés en paramètres, il est alors intégré au corpus produit. Dans le cas où l'on n'utilise qu'une limitation sur la longueur, cela simplifie l'algorithmique sous-jacente. Un simple tampon de *n* mots est employé. Lorsqu'il est plein, c'est-à-dire que le début d'un bloc vérifie le critère de longueur, son contenu est écrit en sortie du filtre. Il est suffisant ensuite de continuer à produire pour chaque nouveau mot en entrée, un mot en sortie s'il fait partie du vocabulaire. Dans le cas contraire, on réinitialise le tampon et on recommence à rechercher un nouveau bloc minimal. Nous allons maintenant illustrer cette technique sur un exemple très simple et pour différentes longueurs de bloc minimal. Le texte suivant est le texte impur en entrée du filtre :

bonjour monsieur durand comment allez vous .

Exemple V.2 : exemple simple de texte impur filtré par blocs minimaux

Si l'on considère que le vocabulaire visé contient tous les mots sauf le nom propre "durand", les résultats de l'extraction de blocs minimaux, en fonction des paramètres fixés, sont les suivants :

Paramètres	**Corpus obtenu (1 bloc minimal par ligne)**
n = 1	<s> bonjour monsieur comment allez vous </s>
n = 2	<s> bonjour monsieur comment allez vous </s>
n = 3	comment allez vous </s>
n = 4 et +	Ø
phrases complètes	Ø

Tableau V.1 : résultat d'extraction de blocs minimaux sur l'exemple V.2

Dans le corpus généré, les balises <s> et </s> représentent respectivement les marques de début et de fin de phrase. Ces marques ne sont pas comptées dans la taille minimale des blocs. Si l'on analyse les résultats précédents on peut voir, même sur ce court exemple, que le réglage des paramètres est à faire avec soin. Le filtre a, de plus, extrapolé un début de phrase au début du texte impur. C'est aussi le cas à chaque début de nouveau document. Sur le tableau V.1, nous constatons que la taille du vocabulaire influe énormément sur le nombre et la taille des blocs minimaux. Ainsi, en augmentant la couverture du vocabulaire, par des mots fortement probables sur la Toile comme nous venons de le voir, le nombre de césures causées par des mots hors vocabulaire diminue, et la taille minimale des blocs augmente.

Dans les méthodes de modélisation statistique du langage, nous avons vu qu'un mot

particulier (<UNK>) permettait d'obtenir des combinaisons qui n'avaient pas été vues lors de la phase d'apprentissage. Dans ce cas, entre chaque césure de bloc, si celle-ci ne provient pas d'une fin de phrase, nous devrions insérer cette balise. Cependant, vu le nombre impressionnant de graphèmes inconnus sur la Toile (plus de 6,5 millions), donc de césures possibles entre des blocs minimaux, la probabilité de cette balise devient disproportionnée par rapport aux mots du vocabulaire. Nous utilisons donc plutôt une définition *a priori* de cette probabilité. Nous la définissons comme étant égale à la probabilité du mot le moins probable. En faisant cela, nous garantissons de pouvoir construire, par méthodes de repli avec pénalité, des bigrammes et trigrammes inconnus. Puisque cette balise est employée uniquement en repli, sa probabilité réelle est donc inférieure à celle du mot le moins probable qui appartient à des trigrammes et des unigrammes connus.

V.2. Influence des paramètres sur la taille du corpus d'apprentissage

Nous avons vu comment étaient extraits les blocs minimaux et quels paramètres ce filtre acceptait. Nous allons maintenant voir quelle est l'influence de ces paramètres sur la taille du corpus d'apprentissage obtenu. Nous ne travaillerons qu'avec un vocabulaire de type CSTAR (2500 mots environs). Ce choix est motivé par le fait que nous utiliserons les mêmes corpus pour l'apprentissage de modèles de langage dans les tests de reconnaissance de la parole du chapitre suivant. Or, comme nous le verrons, ces tests sont très gourmands en ressources système. C'est donc pour garder une cohérence de nos résultats que nous avons limité notre vocabulaire. De plus, à l'époque de ces tests, nous ne possédions pas encore *WebFr4*, nous utiliserons donc *WebFr* pour extraire nos corpus.

En utilisant notre filtre et en faisant varier les paramètres, nous obtenons les résultats qui sont présentés dans le tableau X :

Paramètres	**Nombre de mots dans le corpus**
$n = 1$	280 millions
$n = 2$	200 millions
$n = 3$	145 millions
$n = 4$	71,5 millions
$n = 5$	45 millions
$n = 6$	20 millions
phrases complète de minimum 5 mots	46500

Tableau V.2 : variation de la taille du corpus en fonction des paramètres du filtre

Comme nous pouvons le voir facilement, en augmentant la taille minimale des blocs, on réduit très fortement la taille du corpus obtenu, passant de 280 millions à 20 millions en allongeant la longueur minimale des blocs de 1 à 6. On peut aussi noter une certaine régularité pour des tailles allant de 3 à 6. À chaque fois, dans cet intervalle, en augmentant la taille de 1 on réduit le corpus d'environ 50%. Cela signifie que dans ce cas, pour une longueur *L* donnée, 50% de la masse d'apprentissage est constitué de blocs exactement de taille *L*. Alors, les autres 50% du corpus contiennent des blocs de taille supérieure à *L*. Nous n'avons pas poussé plus loin nos tests ($n > 6$) car pour $n = 6$, la taille du corpus obtenu est du même ordre que ceux habituellement utilisés par les équipes de recherche. L'intérêt quantitatif de notre méthode est alors caduc.

La dernière ligne du tableau X montre aussi un résultat intéressant. Avec l'extraction de phrases complètes de longueur 5, nous n'obtenons que 46500 mots. Dans ce cas, cela devient trop petit pour apprendre des modèles de langage trigrammes de qualité. L'extraction de phrases complètes doit être utilisée avec parcimonie et uniquement avec des vocabulaires très grands, ce qui minimise la césure en blocs minimaux sur des mots hors vocabulaire.

Comme nous l'avons déjà indiqué, nous reprendrons ces corpus et les modèles de langage qui en découlent, pour les évaluer dans le cadre d'une tâche de reconnaissance de la parole, dans le chapitre suivant.

VI. Adaptation des outils de calcul de modèles de langage

VI.1. Problème de l'apprentissage sur des blocs minimaux

La forme du corpus de blocs minimaux que nous obtenons pose un gros problème lors du calcul des probabilités de nos modèles avant même la phase de lissage. Prenons le corpus d'apprentissage suivant, présentant un bloc minimal par ligne, en exemple :

```
<s> bonjour ici monsieur durand je voudrais réserver
<s> c'est une chambre que je voudrais
```

Exemple V.3 : petit corpus de blocs minimaux

Dans ce cas, si l'on cherche à obtenir la probabilité du mot "réserver" connaissant son historique "je voudrais", le résultat de la méthode de calcul habituelle est le nombre de fois que l'on a rencontré le triplet de mots "je voudrais réserver" divisé par le nombre d'occurrences du doublet de mots contenus dans son historique. Dans notre exemple, la probabilité d'avoir "réserver" sachant "je voudrais" est de 0,5.

Le second bloc a été terminé par l'apparition d'un mot inconnu, sinon nous trouverions une marque de fin de phrase. Le résultat précédent serait juste si le "je voudrais" du second bloc avait été utile dans l'apprentissage d'un trigramme, ce qui n'est pas le cas ici puisqu'il est en bout de bloc. Dans l'approche standard des outils de calcul de modèles de langage, il aurait été utile d'utiliser la balise <UNK> pour indiquer la présence d'un mot inconnu à la fin du second bloc. Cependant, comme nous l'avons déjà dit, cela entraînerait une trop forte

probabilité pour le mot inconnu. Dans cette optique, nous pouvons alors considérer que le doublet de mot "je voudrais", à la fin du second bloc, n'apporte aucune information sur le mot qui pourrait suivre.

Le problème provient donc des événements qui se retrouvent à la fin de chaque bloc minimal. Cela est valable pour tous les niveaux d'un modèle n-gramme où $n > 1$, c'est-à-dire où l'historique n'est pas vide. Dans le cas d'un corpus standard d'apprentissage, le même problème n'apparaît qu'une fois, en fin du corpus et c'est pour cela qu'il est négligé.

VI.2. Modifications du calcul des probabilités

Nous devons modifier le recensement des événements observés dans notre corpus d'apprentissage pour obtenir des calculs justes sur la capacité de prédiction d'une séquence de mots. Pour cela, définissons deux compteurs au lieu du seul présent dans les techniques habituelles de calcul de modèles de langage. Le premier, appelé N_1, nous servira à dénombrer les événements et correspondra au compteur habituel. Le second, N_2, servira à compter combien de fois un événement a été observé comme historique d'un autre. Évidemment, dans un modèle n-gramme, dans le cas du niveau n, N_2 vaut toujours 0. En effet, dans le cas de $n = 3$, aucun trigramme ne peut être l'historique d'un autre événement qui serait alors de longueur 4. Les formules pour le calcul des probabilités d'un mot sachant son historique (voir chapitre III section I.2) sont alors transformées.

$$P(\mathrm{m} \mid h) = \frac{N_1(h, \mathrm{m})}{N_2(h)}$$

Équation V.1 : calcul de la probabilité de la séquence de mots h, m avec des blocs minimaux

En réalisant cette modification, nous obtenons l'effet voulu, nous ne comptabilisons les événements en contexte que dans l'éventualité où ils ont participé à la construction d'un niveau de n-gramme supérieur. Si l'on reprend l'exemple V.3, la probabilité d'obtenir "réserver" sachant "je voudrais" devient égale à 1. C'est exactement ce que reflète le corpus d'apprentissage puisque "réserver" n'a été trouvé que comme successeur de "je voudrais" et que ces deux mots n'ont jamais été vus dans d'autres contextes de trigrammes.

L'autre avantage de cette méthode de comptage est qu'elle reste tout à fait compatible avec des approches de lissage et redistribution comme celle de Katz [Katz 87] par exemple (voir chapitre III section II.2).

Conclusion

Nous avons détaillé notre méthode pour la construction automatique de modèles de langage statistiques en utilisant les documents en provenance d'Internet. Nous avons mis au point un filtrage adapté à tout type de texte impur, c'est-à-dire contenant virtuellement plus de vocabulaire que l'application visée, qui se nomme filtre par blocs minimaux. Enfin, nous avons modifié les formules de calcul des probabilités conditionnelles d'apparition d'un mot

en fonction de son historique pour tenir compte des spécificités d'un corpus de blocs minimaux. Tout cela nous permet de construire facilement de nouveaux modèles de langage en ne définissant que le vocabulaire de base de l'application [Vaufreydaz et al. 01a].

Le chapitre suivant présente diverses expérimentations et les résultats que nous obtenons, sur diverses tâches de reconnaissance, en utilisant nos techniques automatiques.

Chapitre VI : Expérimentations et résultats

Chapitre VI : Expérimentations et résultats

Présentation du chapitre

Ce chapitre présente les résultats de l'utilisation de la méthode de filtrage de documents en provenance d'Internet. Dans une première partie, nous décrirons le système de reconnaissance utilisé lors de nos tests. Nous préciserons ensuite les mesures employées dans nos évaluations. Nous présenterons alors une étude statistique de nos modèles de langage et l'apport des documents d'Internet. Nous terminerons par l'étude de nos outils dans diverses tâches de reconnaissance allant de quelques milliers à plus de 20000 mots.

I. Raphaël, le système d'expérimentation

I.1. Présentation

RAPHAEL [Akbar et al. 98] (Reconnaissance Automatique de PHrases, d'Acronymes et d'Expressions Langagières) est le système de reconnaissance de la parole spontanée du laboratoire CLIPS. Il a été entièrement défini et élaboré par l'équipe GEOD. Il est basé sur la boîte à outils Janus développée par CMU, cependant l'apprentissage acoustique est entièrement réalisé par nos soins et les modèles de langage sont ceux que nous présentons dans cette thèse.

I.2. Description technique

I.2.a. Boîte à outils Janus-III

Janus est une boîte à outils dédiée à la traduction de parole [Waibel et al. 91]. Nous n'utilisons pour notre part que la partie reconnaissance de la parole. Janus possède toutes les briques de base pour la définition d'un système de reconnaissance : module d'extraction de

paramètres acoustiques, module gérant les HMMs, différents algorithmes de recherche, etc. Janus est une extension du langage de scripts Tcl/Tk [Welch 95]. Cela donne l'avantage d'obtenir la portabilité des scripts qui décrivent les systèmes. Cependant, le code source de Janus-III n'était, lui, pas portable. Nous avons réalisé nous-même son portage sous Windows™ de Microsoft, environnement de travail au sein de notre équipe de recherche.

I.2.b. Modèles acoustiques

Nous utilisons comme paramètres acoustiques les 13 premiers coefficients MFCCs et l'énergie. Nous calculons les dérivées première et seconde de ces coefficients. Nous ajoutons à cela le *zero-crossing rate* (Janus n'incluant pas le *band-crossing*) pour obtenir pour chaque fenêtre d'analyse un ensemble de 43 paramètres. Ceux-ci sont extraits toutes les 10 millisecondes sur une fenêtre de 20 ms, ce qui signifie que le recouvrement entre deux fenêtres successives est de 10 ms. Avant le calcul de nos paramètres, une fenêtre de Hamming [Bellanger 95] est appliquée. Enfin, pour réduire l'espace de représentation, nous appliquons une LDA (*Linear Discriminant Analysis*) pour obtenir 24 coefficients. Pour résumer, nous obtenons donc toutes les 10 ms un jeu de 24 coefficients représentant les paramètres extraits du signal. Le lecteur se référera au chapitre II « Reconnaissance de la parole » pour la description de ces coefficients.

Les modèles acoustiques sont appris sur le corpus sonore BREF-80 [Lamel et al. 01] qui contient environ 10 heures de parole. Raphaël fonctionne sur une base de 46 unités acoustiques qui sont les phonèmes du français, leurs variantes, 'o' ouvert ou fermé par exemple, plus une unité acoustique spéciale pour modéliser le silence. Ces unités sont, pour l'instant, traitées hors contexte. On ne différencie pas les phonèmes en fonction des phonèmes qui les précèdent ou qui les suivent. Toutes les unités sont construites sur une topologie de HMMs dites de Bakis, c'est-à-dire un HMM gauche-droit d'ordre 1 à 3 états, sauf le silence qui lui compte 5 états. Nous travaillons en parallèle à la rédaction de ce manuscrit, à la mise au point d'un système modélisant les unités acoustiques en triphones. Nous utilisons pour cela BREF-TOTAL qui est un corpus bien plus gros que BREF-80. Les résultats présentés dans cette thèse sont obtenus avec le modèle de phonèmes indépendants du contexte appris sur BREF-80.

I.2.c. Modèles de langage

Bien que Janus soit tout à fait capable d'intégrer des grammaires, nous n'utilisons que des modèles de langage trigrammes pour les raisons que nous avons énoncées plusieurs fois depuis le début de ce manuscrit. Ceux-ci sont appris sur nos corpus en provenance d'Internet avec les méthodes décrites au chapitre précédent. Nous utilisons le lissage de Katz [Katz 87]. Les sections suivantes de ce chapitre présentent les résultats que nous obtenons dans ces conditions.

I.2.d. Algorithmes de recherche

Les algorithmes de recherche sont fondés sur une représentation arborescente du dictionnaire acoustique. C'est ce que [Woszczyna 98] présente sous le nom de recherche *Tree-Forward*. Une fois la phase de construction de l'arbre terminé, un treillis de mots est

construit (cf. chapitre II). Une nouvelle passe, n'utilisant cette fois que le modèle de langage, fournit l'hypothèse finale de reconnaissance.

I.3. Intégration aux démonstrateurs des projets CSTAR et Nespole!

Nous avons intégré RAPHAEL dans deux démonstrateurs des projets européens dont le laboratoire. Pour le projet CSTAR [Web 01], le système de reconnaissance de la parole fonctionnait localement, c'est-à-dire que le système de visioconférence n'était là que pour réaliser le face à face entre les deux utilisateurs du système. Il n'y avait donc aucun transfert de données parole pour faire de la reconnaissance sur un site distant. L'architecture était donc une reconnaissance faite sur une machine dédiée sur le réseau local. Pour cela, nous avons défini un protocole propriétaire de transfert de données et de récupération de résultats décrit dans [Vaufreydaz et al. 99a]. Le système de synthèse de la parole, donc du texte au signal sonore, fonctionnait sur le même principe. Ce système a été utilisé avec succès lors de la démonstration finale du projet CSTAR, dans sa phase II, en juillet 1999 [Presse 99].

Pour ce qui est du projet Nespole!, le principe de fonctionnement est différent. En effet, le système est sur un site distant et utilise directement le signal de parole encodé par le système de visioconférence [Besacier et al. 01a], via le protocole H323 [Black 00] pour la reconnaissance. L'architecture globale du système est composée d'une grappe de serveurs de traduction (reconnaissance, traduction et synthèse) qui sont appelés par un logiciel intermédiaire appelé le *médiateur*. Les données transitent du logiciel de visioconférence du client vers le médiateur. Elles sont ensuite traitées par le serveur adéquat puis transférée vers l'agent, et inversement. Cela permet l'intégration de ce système avec tous les logiciels de visioconférence du marché respectant la norme H.323. Comme nous l'avons déjà indiqué, la difficulté de ce mode de fonctionnement, contrairement à CSTAR, est non seulement la dégradation du signal de parole due au codage mais aussi à la perte de données provoquée par l'utilisation du protocole UDP/IP. Nous avons réalisé plusieurs études pour prendre en compte ces données dans l'apprentissage de nos modèles acoustiques [Bergamini 00], [Besacier et al. 01b], [Besacier et al.], [Lamy 01].

II. Mesures utilisées

Pour juger de la pertinence et de la qualité de nos modèles de langage, nous avons besoin d'utiliser différentes mesures [Chen et al. 98]. Nous allons décrire dans cette section celles que nous emploierons dans nos différents tests.

II.1. Théorie de l'information

II.1.a. Entropie

Dans la théorie de l'information, un langage L est considéré comme une source produisant une suite de mots m_i à partir d'un ensemble fini d'élément V, le vocabulaire [Abramson 63]. L'entropie, souvent notée $H(L)$, est la quantité d'information portée par chacun des éléments. Le nom *entropie* est tiré de la mesure de désordre en thermodynamique car la formule de calcul de ces deux entités est mathématiquement similaire. Dans le cadre du

langage, la quantité d'information, en base *b*, contenue dans une séquence de mots $S = m_1...m_n$ vaut $-\log_b P(S)$, où P est la vraisemblance de la séquence S. Intuitivement, plus une suite de mots est improbable, plus sa quantité d'information augmente. Il est maintenant possible de calculer l'entropie avec l'équation VI.1.

$$H_b(S) = -\sum_{m_i} P(m_i)\log_b P(S)$$

Équation VI.1 : Formule de calcul de l'entropie d'une séquence S

Dans cette équation, $P(m_i)$ est la probabilité d'émission de m_i, le *i-ème* mot de la séquence. Alors, d'après le théorème de Shannon [Shannon 48], tout codage d'une séquence *S*, dans une base *b*, nécessite en moyenne pour être codé $H_b(S)$ chiffres. À partir de maintenant, nous simplifierons notre notation en utilisant H pour H_2. Cela correspond au codage de l'information en base 2, l'une des bases utilisées en informatique.

Le problème de cette formule réside dans l'estimation de l'entropie d'un langage complet. En effet, pour obtenir la valeur la plus fiable possible de l'entropie, il faut pouvoir observer le plus de séquences possibles. La valeur de l'entropie d'un langage *L* peut alors se calculer par la formule suivante, en sommant sur un nombre maximum de séquences possibles *N*, tendant vers l'infini :

$$H(L) = -\lim_{N \to \infty} \frac{1}{N} \left(\sum P(m_1...m_n)\log P(m_1...m_n) \right)$$

Équation VI.2 : formule de calcul de l'entropie d'un langage L

Ce calcul se révèle infaisable car il existe une infinité de séquences possibles. On fait alors un calcul approché en estimant que si l'on possède une séquence suffisamment longue, elle est dite ergodique, c'est-à-dire qu'elle représente toutes les possibilités qu'offre le langage. Dans ces conditions, il est suffisant de calculer l'entropie de cette source ergodique pour avoir celle du langage.

II.1.b. Perplexité

La perplexité est la mesure la plus couramment employée depuis de nombreuses années [Jelinek et al. 77] pour juger de la qualité d'un modèle de langage. Cette valeur se calcule sur un texte non vu au cour de l'apprentissage. C'est ce que l'on nomme en anglais *test-set perplexity* [Rosenfeld 94], à ne pas confondre avec la *training-set perplexity* qui est la perplexité du modèle sur ses données d'apprentissage. Elle est facteur de l'entropie $H(M)$ d'un modèle *M* utilisé dans la prédiction des mots rencontrés. Elle peut être obtenue de la manière suivante :

$$PP(M) = 2^{H(M)}$$

Équation VI.3 : formule de la perplexité d'un modèle M

Dans le cadre de l'emploi d'un modèle de langage statistique, le calcul de l'entropie de ce modèle se fait au moyen d'une quantité nommée *logprob* [Jelinek 89]. Celle-ci est définie comme suit pour une séquence de mots *S* de longueur *N*.

$$logprob(S) = -\frac{1}{n} \log \left(\prod_{i=2}^{n} P(m_i | h) \right)$$

Équation VI.4 : formule de la logprob d'une séquence S

La formule de la perplexité se réduit alors par simplification, dans le cas du calcul de la perplexité d'un modèle statistique sur un texte de test représenté par une séquence *S*, à

$$PP(M) = 2^{logprob(S)} = \left(\prod_{i=2}^{n} P(m_i | h) \right)^{-1/n}$$

Équation VI.5 : calcul de la perplexité d'un modèle statistique

La perplexité peut aussi être vue comme le facteur de branchement moyen du langage considéré. Si l'on prend un modèle de langage ayant une perplexité *X*, il est équivalent, en terme de perplexité, à un langage qui comporterait *X* mots équiprobables. Alors, un système de reconnaissance de la parole utilisant un modèle de langage de perplexité *X* peut se contenter, en moyenne, de n'évaluer que les *X* hypothèses les plus probables à chaque instant.

Une méthode alternative de calcul de la perplexité a été développée pour permettre une plus grande justesse de résultats [Bimbot et al. 97], [Jardino 98], [Bimbot et al.]. En effet, si l'on regarde les équations précédentes, il est évident que leur résultat est très dépendant du modèle et du texte sur lequel on l'évalue. La variante proposée est basée sur le jeu de Shannon [Shannon 51]. Ce jeu, mis au point dans les années 50 par C. Shannon, avait pour but l'estimation de l'entropie d'un langage. Il était demandé à des sujets de prédire la lettre suivante en connaissant ou non les lettres précédentes. En réitérant les expériences avec différents sujets et plusieurs historiques, il est possible de déduire quelle quantité d'information est présente dans le langage considéré.

La variante de calcul de la perplexité utilise le même type de procédé. L'idée est de construire un ensemble de test constitué de phrases tronquées dont le contenu linguistique est complètement différent. On demande c'est-à-dire de donner une probabilité avec un modèle de langage, sur le mot manquant à la fin de la séquence et seulement lui. La perplexité est alors calculée à partir de ces mises. La difficulté de cette méthode réside dans la définition du corpus de test pour être certains de tester tous les phénomènes présents dans le langage considéré. Cependant, d'après les résultats présentés dans [Bimbot et al.], cette méthode semble prometteuse.

II.1.c. Autres approches

Bien qu'elle soit utilisée depuis très longtemps, la perplexité a montré son imperfection, pour la reconnaissance de la parole, au cours des expérimentations de diverses équipes de recherche. Selon [Ito et al. 99], le facteur de corrélation entre la perplexité et le

taux de reconnaissance n'est que de 0,59. D'autres approches ont donc été utilisées pour tenter de mesurer l'adéquation d'un modèle de langage à une tâche de reconnaissance de la parole.

La perplexité acoustique, intégrant les transcriptions acoustiques des mots du modèle, a été suggérée mais [Jelinek 89] a montré qu'elle était simplement proportionnelle à la perplexité. [Ferretti et al. 89] proposaient de combiner l'information acoustique et linguistique par le biais de l'entropie du système de reconnaissance (*Speech Decoder Entropy*). [Chen et al. 98] ont proposé deux approches. L'une, nommée *M-ref*, se calcule en fonction de la fonction de l'erreur de vraisemblance des mots appris. La seconde est basée sur une simulation d'un processus de reconnaissance. [Ito et al. 99] indiquent que ces deux propositions n'ont pas une très forte corrélation avec le taux de reconnaissance. Leur proposition est d'utiliser une différence normalisée entre le score du mot et le score du meilleur mot, sachant le contexte courant, avec des fonctions de seuillage de type sigmoïde pour l'obtention du résultat final. Le facteur de corrélation entre cette métrique et le taux de reconnaissance est supérieur à 0,8, et va jusqu'à 0,85 dans certaines expérimentations.

II.2. Taux de reconnaissance

II.2.a. Distance de Levenshtein et de Damerau-Levenshtein

La distance de Levenshtein [Levenshtein 66] ou distance d'édition est couramment utilisée dans de nombreuses applications où il faut mesurer la similarité entre deux séquences : la gestion de versions de fichiers (la commande *diff* sous Unix par exemple), la génétique avec l'alignement de séquences d'ADN, dans certains systèmes de correction d'erreurs orthographiques, lors du choix du mot le plus proche et enfin la mesure de performance en reconnaissance de la parole [Sankoff et al. 83]. Elle permet, à partir d'un alphabet et de deux séquences, de déterminer quelle est la longueur d'une séquence minimale d'opérations pour transformer la première séquence en la seconde. Ces opérations sont au nombre de trois et portent sur les lettres de l'alphabet considéré :

- substitution d'une lettre par une autre (`S`)
- omission d'une lettre (`O`)
- insertion d'une lettre (que nous noterons `I`)

La distance dite de Damerau-Levenshtein [Damerau 64], [Levenshtein 66] (souvent noté DLM dans la littérature pour *Damerau-Levenshtein Metric*, nous utiliserons la version française DDL) ne considère plus simplement chaque erreur comme possédant un coût unitaire mais affecte des poids à chaque opération. Elle s'écrit ainsi pour une distance entre deux séquences S_1 et S_2 :

$$DDL(S_2, S_1) = DDL(S_1 \rightarrow S_2) = \text{argmin}(N_S x P_S + N_O x P_O + N_I x P_I)$$

Équation VI.6 : Distance de Damerau-Levenshtein

N_S, N_O et N_I sont respectivement le nombre de substitutions, d'omissions et de d'insertions et P_S, P_O et P_I le poids positif ou nul associé à chacune de ces opérations. Wagner et Fischer [Wagner et al. 74] ont montré que la minimisation de la DDL peut s'exprimer sous la forme d'une récurrence et ont proposé un algorithme de résolution par

programmation dynamique. Le lecteur pourra trouver des informations sur l'utilisation de cette mesure de distance pour l'apprentissage et le test de modèles de langage syntaxiques stochastiques dans le cadre de dialogues dans [Miclet et al. 99].

Dans le cadre où nous l'utiliserons, c'est-à-dire l'évaluation de système de reconnaissance, chaque élément de notre alphabet est un mot et le calcul de cette distance repose sur deux séquences. La première, nommé référence (REF), représente la transcription corrigée de ce qui est énoncé dans la portion de signal à reconnaître. La seconde, l'hypothèse (HYP), correspond à la solution la plus probable trouvée par le système. Pour que les deux soient comparables, les mêmes règles de transcription doivent être appliquées à ces deux éléments. Nous détaillerons plus loin quelles sont les règles que nous avons suivies. L'exemple suivant nous donne un exemple d'alignement réalisé avec cette métrique :

```
REF: bon     c' est d' accord pour les nuits de  lundi neuf et   mardi dix
HYP: comment c' est d' accord tous les nuits *** lundi neuf août mardi dix août
ERR:   S                        S             O             S               I
```

Exemple VI.1 : Alignement entre une phrase de référence et une hypothèse de reconnaissance

Dans cet exemple, nous voyons qu'il existe 3 types d'erreur. Les substitutions, notées 'S', correspondent aux mots substitués à d'autres. Le 'O' correspond à une omission, c'est-à-dire à un mot qui n'a pas été trouvé par le système, c'est généralement le cas des mots courts prononcés trop rapidement. Enfin, il arrive parfois que des mots soient insérés par erreur. Ces derniers sont notés 'I'. L'alignement proposé ci-dessus comprend donc 10 mots correctement reconnus et 5 erreurs, pour un total de mots à reconnaître de 14. Des procédures d'alignement plus performantes pour limiter le cumul des erreurs ont été décrites dans [Fisher et al. 93].

II.2.b. Taux de Mots Corrects (TMC)

Le Taux de Mots Corrects (*Word Correct Rate* dans la littérature en anglais) est la mesure la plus intuitive pour mesurer les performances d'un système de reconnaissance de la parole. Il est basé sur une transcription correcte des éléments à reconnaître, la référence, et l'hypothèse de reconnaissance. Un alignement entre la référence et l'hypothèse est généré et permet de comptabiliser combien de mots ont été correctement reconnus. L'équation de ce taux est donc :

$$\text{Taux de Mots Corrects} = \frac{\text{nombre de mots correctement reconnus}}{\text{nombre de mots à reconnaître}}$$

Équation VI.7 : Calcul du Taux de Mots Corrects (TMC)

Si l'on considère l'exemple précédent, le TMC est égal à :

$$\text{TMC} = \frac{10}{14} \approx 71\%$$

Le problème majeur de ce calcul est qu'il ne prend pas en compte toutes les erreurs commises. Si l'on modifie l'exemple précédent comme suit :

```
REF: bon c' est d' accord pour les nuits de lundi neuf et mardi dix
HYP: bon c' est d' accord pour les nuits de lundi neuf et mardi dix août
ERR:                                                               I
```

Exemple VI.2 : Mise en évidence de l'inadéquation du TMC

Dans ce cas, malgré l'erreur, le TMC = 100%. C'est pour cette raison que dans la majorité des évaluations, une autre mesure est utilisée : le taux d'erreur et son dual, le taux de reconnaissance.

II.2.c. Taux d'erreurs et taux de reconnaissance

L'utilisation du taux d'erreur (*Word Error Rate* en anglais) pour l'évaluation des performances résout les problèmes d'incohérence que nous venons d'évoquer. Ensuite en déduisant le taux d'erreur, nous pouvons calculer le taux de reconnaissance (*Word Accurracy*) qui est une mesure plus juste que le simple TMC. Nous obtenons l'ensemble d'équations suivant :

$$\text{Taux d'erreur} = \frac{\text{nombre de substitutions} + \text{nombre de insertions} + \text{nombre de suppressions}}{\text{nombres de mots à reconnaître}}$$

Équation VI.8 : Calcul du Taux d'Erreurs

$$\text{Taux de reconnaissance} = 1 - \text{Taux d'erreur}$$

Équation VI.9 : Taux de reconnaissance

Si l'on reprend le cas de notre premier exemple, nous obtenons les résultats suivants :

$$\text{Taux de reconnaissance} = 1 - \frac{5}{14} \approx 64\%$$

Ce résultat, plus sévère que le TMC (64% contre 71% précédemment), permet la mise en évidence de toutes les erreurs du système de reconnaissance. Pourtant cette mesure, plus juste, possède aussi un revers. En effet, dans des tâches telles que la reconnaissance d'enregistrements continus avec de grandes portions de silence, si la détection de silence n'est pas optimale, il est évident on aura beaucoup d'erreurs d'insertion, et, dans les cas extrêmes, plus d'insertions que de mots à reconnaître. Dans ces conditions, le taux d'erreur serait donc supérieur à 1 et le taux de reconnaissance inférieur à 0.

II.3. Définition d'une métrique : l'information contextuelle

Dans le cas d'un apprentissage sur un corpus textuel "linéaire", c'est-à-dire non formé de blocs minimaux, on sait pour tous les mots comment ils sont prédits par leurs prédécesseurs et comment ils prédisent leurs successeurs. Or, au sein d'un corpus d'apprentissage formé de blocs minimaux, ce n'est pas le cas car nous avons un problème d'apprentissage sur toutes les césures entre les blocs minimaux. Ce problème de césure n'est présent dans un corpus linéaire qu'au début et à la fin du corpus qui peut être considéré comme un seul bloc minimal.

Pour mesurer quelle quantité d'information apporte un mot à notre modèle de langage, nous avons défini la notion d'information contextuelle portée par un mot au sein d'une séquence utilisée pour l'apprentissage d'un modèle n-grammes. En considérant uniquement la position du mot au sein de la séquence, et non pas le mot lui-même, cette mesure identifie le nombre de mots qui se retrouve, par le mécanisme d'apprentissage, dans toutes les positions des contextes d'un n-grammes. Nous noterons cette valeur IC_n pour Information Contextuelle en contexte de longueur *n*. Son calcul est décrit ci-dessous :

$$IC_n(\mathrm{S}) = 0 \text{ si la longueur de S} < (2.n - 1)$$
$$IC_n(\mathrm{S}) = \text{longueur de S} - 2 \times (n - 1) \text{ sinon}$$

Équation VI.10 : calcul de l'information contextuelle

Par exemple, dans le cas d'une séquence de 3 mots et pour un modèle bigramme, le second porte une information contextuelle complète puisque l'on apprend comment le prédire avec le premier et comment il prédit le troisième. Dans ce cas, $IC_2(S)$ est bien égale à 1 puisque seul le second mot de la séquence comporte une information contextuelle complète pour un modèle bigramme. Cette métrique n'est en aucun cas utile dans le cas de n=1. Effectivement, puisque nous comptons alors tous les unigrammes, l'information contextuelle est donc égale à la longueur de la séquence *S*. Évidemment, cette mesure vaut aussi avec d'autres types d'éléments, par exemple pour les modèles n-classes, n-Seq, etc.

II.4. Choix

Le choix parmi ces mesures avait plusieurs objectifs. Premièrement, il devait permettre d'avoir des métriques utilisées par d'autres, fournissant alors des points de comparaison avec d'autres travaux de recherche. De plus, il devait apporter une information pertinente et juste.

Nous avons choisi d'utiliser la perplexité comme mesure pour les modèles de langage. Même si d'autres mesures présentent un intérêt, par exemple la mesure décrite par [Ito et al. 99], nous ne fournirions aucune information sur nos modèles car nous n'avons trouvé aucun résultat sur le français obtenu ces autres mesures. Ensuite, pour des raisons évidentes de justesse des résultats affichés, nous utiliserons le taux de reconnaissance et non pas le taux de mots corrects.

Il faut noter ici que nous avons employé deux outils différents au cours de nos différentes expérimentations. Le premier, nommé *align*, est fourni avec le toolkit Janus III [Waibel et al. 91]. Le second est *sclite*, un outil fourni par le NIST [Web 03] pour ses évaluations. Avec les mêmes règles de transcription des hypothèses et des références, ces deux outils fournissent bien entendu le même résultat. Ayant, commencé les calculs avec *align* dans certaines conditions expérimentales, et possédant par ailleurs les résultats d'autres laboratoires obtenus avec *sclite* pour d'autres expériences, nous avons cependant choisi de ne pas interchanger les logiciels dans nos évaluations, pour être certain d'avoir des résultats comparables.

III. Études statistiques

III.1. Introduction

Les statistiques que nous donner maintenant concernent les modèles trigrammes que l'on peut obtenir avec le corpus *WebFr* et la répartition de l'information au sein de ce corpus. Ces expérimentations étant très gourmandes en temps de calcul et en capacité disque pour stocker plusieurs milliers de modèles de langage, nous n'avons travaillé que sur la tâche de réservation touristique type CSTAR. Celle-ci se résume à un petit vocabulaire d'environ 2500 mots, ce qui nous permet d'accélérer les phases de test et de limiter la taille des modèles trigrammes que nous calculons.

Le corpus textuel extrait par la méthode des blocs minimaux sur *WebFr* dans ce cadre compte, comme nous l'avons déjà vu, presque 145 millions de mots avec une taille minimale des blocs égale à 3. Nous n'utiliserons que les 46 premiers millions de mots pour nos expériences, celles-ci s'avérant trop longue à conduire sur le corpus entier. Nous avons donc découpé ce corpus en morceaux d'environ 2 millions de mots. La césure entre deux morceaux est faite entre deux blocs minimaux. Nous obtenons un ensemble de 23 morceaux, et pouvons ainsi mesurer l'influence de chaque partie de notre corpus.

III.2. Répartition de l'information au sein de *WebFr*

Nous allons nous attacher à mesurer si la répartition de l'information est uniforme sur notre corpus. Notre première intuition est que cela est forcément le cas puisque le robot *Clips-index* ne possède pas de stratégie de collecte particulière. Seuls la vitesse de réponse des serveurs Web et le temps de transfert sur le réseau Internet conditionnent l'ordre de collecte des documents.

Pour valider cette hypothèse, nous avons conduit un H-Test qui consiste à utiliser tous les éléments à étudier sauf un seul qui est mis à part. Il suffit ensuite de refaire les mêmes statistiques en enlevant successivement chaque élément et de calculer la moyenne et l'écart type entre chaque test pour évaluer l'importance relative de chaque élément. Dans notre cas, chaque élément considéré est un bloc de 2 millions de mots de *WebFr*. Nous nous intéressons dans cette expérience au nombre de contextes trigrammes connus par le modèle de langage. Cela nous donne une information sur la couverture linguistique de chaque bloc en nombre de contextes différents. Si l'un des blocs à une plus grande importance que les autres sur la variété des contextes connus, nous pourrons le détecter.

Nous nous retrouvons avec 23 corpus de test comprenant chacun environ 44 millions de mots. Pour chacun, nous avons calculé un nouveau modèle de langage tous les 20000 mots d'apprentissage pour mesurer le nombre de contextes connus. Cela nous permet d'otenir une courbe moins lissée que si l'on avait pris simplement les blocs complets. Pour pouvoir visualiser les trigrammes présents dans chacun de ceux-ci et faire quelques tests de reconnaissance, nous avons dû sauvegarder chacun des modèles obtenus. Pour chaque corpus de test, nous avons calculé 2286 modèles de langage, soit au total 52578 modèles. Ensuite,

pour chaque modèle utilisant *x* mots pour l'apprentissage, *x* allant de 20000 à plus de 44 millions, nous calculons la moyenne du nombre de contextes trigrammes connus, et l'écart type sur les 23 corpus. Les résultats sont présentés sur le graphique suivant.

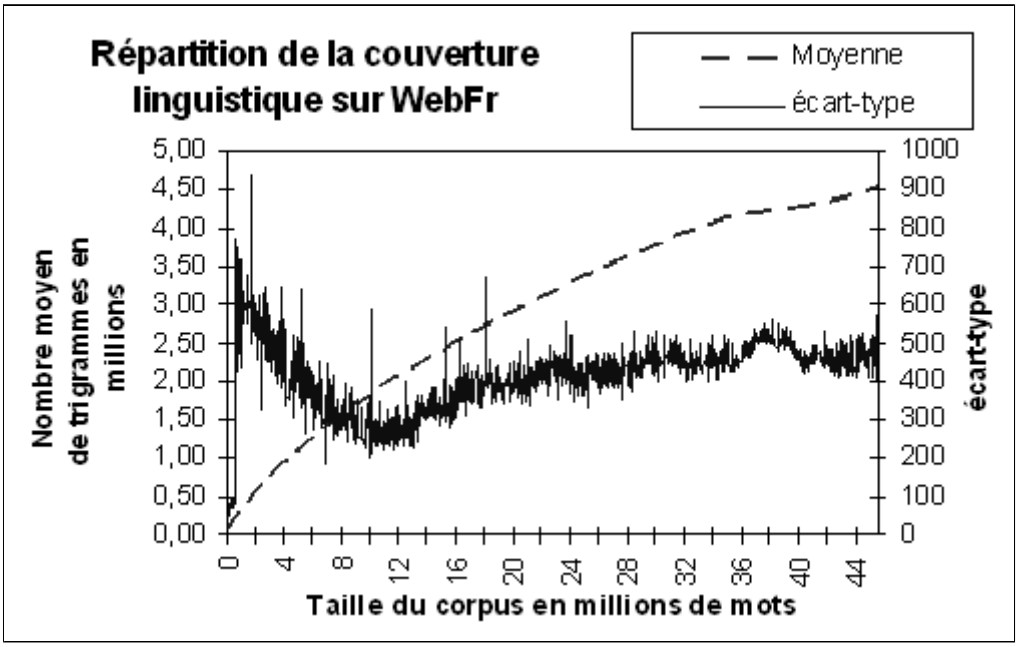


Figure VI.1 : répartition de la couverture linguistique sur WebFr

Nous pouvons remarquer que la courbe de la moyenne présente une évolution constante, hormis sur la fin où l'accroissement du nombre de contextes semble s'atténuer. Cela peut-être expliqué par l'obtention d'une couverture presque maximale avec 34 millions de mots. Pourtant, vue la grande taille du corpus, il nous est difficile d'être formel à ce sujet. Nous verrons, dans les deux sections suivantes, des tests sur la perplexité réalisés dans les mêmes conditions qui nous apporterons plus d'informations sur ce phénomène. Le résultat le plus intéressant de ce graphique est le très faible écart-type entre les résultats. Il ne dépasse jamais 1000 alors que nous travaillons avec des valeurs de l'ordre du million. De plus, sur les tests effectués sur de très gros corpus, cette valeur se situe autour de 500.

Nous pouvons conclure, après examen de ces résultats, que toutes les parties de *WebFr* apportent la même quantité d'information au modèle de langage. Cependant, la totalité du corpus comprend plus de données que chacune des sous-parties. Alors, nous pouvons affirmer que l'information est répartie uniformément sur notre corpus d'apprentissage. L'emploi du corpus complet pour l'apprentissage de nos modèles est donc obligatoire pour maximiser la couverture en nombre de contextes trigrammes de notre modèle.

III.3. Étude de la perplexité

La perplexité est une mesure très répandue pour l'évaluation des modèles de langage. Nous avons effectué les calculs de la perplexité de nos modèles de langage sur le corpus textuel de réservation touristique, puisqu'il contient le type d'énoncés que nous nous efforçons de modéliser. Nous avons employé le découpage en 23 morceaux de *WebFr*. Nous obtenons donc 23 modèles de langage appris progressivement avec 2 millions de mots, puis 4, etc. Comme nous avons montré précédemment que tous les blocs avaient la même

importance, l'ordre de calcul sur ces blocs n'est pas crucial et ne biaise pas nos résultats Le graphique suivant présente l'évolution de la perplexité et du nombre de trigrammes connus dans le modèle de langage.

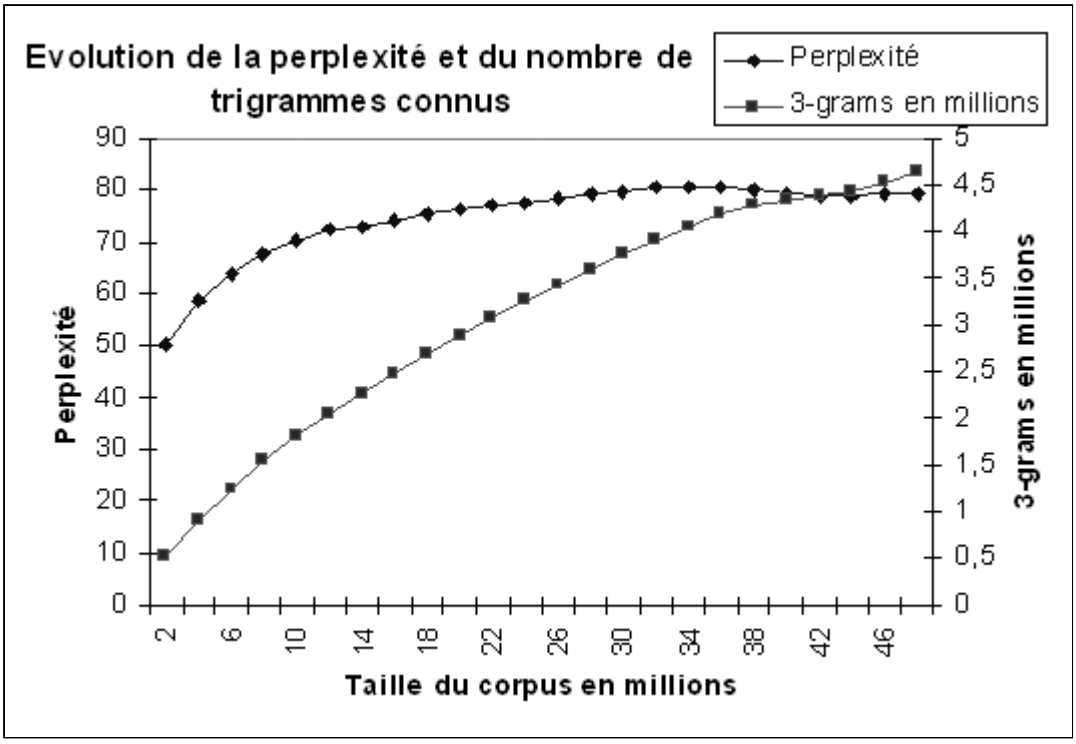


Figure VI.2 : évolution de la perplexité et du nombre de trigrammes connus

La première remarque que nous pouvons faire est que la courbe du nombre de trigrammes est similaire à celle obtenue lors de la précédente expérience. À partir de 38 millions de mots en apprentissage, la pente s'atténue aussi. Nous pouvons dire que ce phénomène n'est pas directement corrélé avec la perplexité du modèle. En effet, celle-ci commence à osciller faiblement autour de la valeur 80 avec un modèle calculé sur seulement 28 millions de mots. Cette valeur est d'ailleurs tout à fait raisonnable au regard de la taille de notre vocabulaire.

Nous pouvons conclure sur ce graphique que, malgré l'arrivée de nouveaux trigrammes au sein du modèle, la perplexité n'évolue plus. Cela tendrait à prouver que dans les 28 premiers millions de mots du corpus d'apprentissage, ou dans 28 millions de mots pris au hasard dans celui-ci, l'information étant uniformément répartie, nous trouvons la majorité de la couverture langagière de notre tâche de réservation touristique. Le surplus de corpus sert donc à affiner les probabilités, l'apparition de nouveaux trigrammes nous donne la variété que nous recherchons pour nos modèles.

III.4. Couverture du langage en trigrammes

Le but d'un modèle de langage est de représenter le langage avec des trigrammes et de ne pas avoir à employer des méthodes de repli avec des modèles d'ordre inférieur pour prédire les mots du corpus de test. Nous pouvons utiliser le pourcentage de mots prédits avec des trigrammes, des bigrammes et des unigrammes lors du calcul de la perplexité pour

mesurer la couverture langagière. Effectivement, nous avons jusqu'à maintenant considéré que l'augmentation du nombre de trigrammes dans le modèle était suffisante pour juger de cette couverture. Pourtant, il est possible, de par la nature des corpus que nous utilisons, que l'augmentation du nombre de trigrammes ne le soit que par des contextes représentant des erreurs présentes sur Internet et ne soit pas due à un phénomène langagier réel.

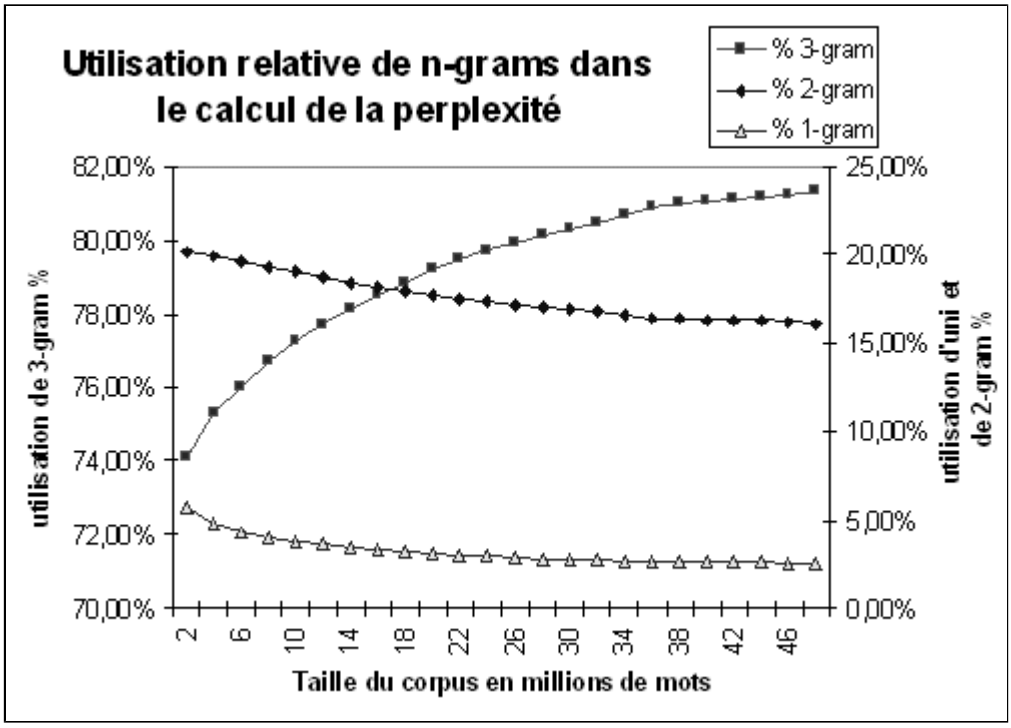


Figure VI.3 : utilisation relative de n-grams dans le calcul de la perplexité

Le graphique de la figure VI.3 précédent montre l'évolution du pourcentage de mots prédits avec des trigrammes (axe de gauche), des bigrammes et des unigrammes (axe de droite). Tout d'abord, nous voyons qu'avec seulement 2 millions de mots, nous prédisons déjà 74% des mots, du corpus de test, et ce avec des trigrammes. Encore une fois, cela corrobore ce que nous énoncions auparavant dans ce manuscrit à savoir que les éléments de base du français sont présents dans tous nos blocs. Par la suite, le nombre de trigrammes connus augmentant, ce pourcentage ne cesse de croître pour atteindre 81% à la fin. Dans le même temps, les pourcentages prédits avec des bigrammes et des unigrammes passent respectivement de 20% à 16% et de 5 à 2%.

Bien que la valeur de la perplexité puisse être considérée comme constante après 28 millions de mots, l'augmentation du nombre de trigrammes permet une prédiction de meilleure qualité, c'est-à-dire avec moins de replis. Cela indique que le modèle de langage est donc plus performant, car plus couvrant, tout en gardant une perplexité identique. Cela justifie les critiques que d'autres équipes de recherche, citées plus haut, et nous même avons émises à propos de l'utilité de la perplexité en reconnaissance de la parole.

Notre expérience montre qu'elle reste cependant un indice intéressant qui peut s'avérer utile en reconnaissance de la parole. Dans des approches de type *Beam Search* (voir chapitre

I), le réglage des seuils empiriques d'élagage des hypothèses en cours peut être fait en fonction de la valeur de perplexité du modèle de langage. La taille de l'espace de recherche peut ainsi être contrôlée.

IV. Application à la reconnaissance de la parole

IV.1. Description des corpus sonores utilisés

Les corpus sonores sur lesquels nous allons évaluer nos méthodes automatiques de construction de modèles de langage sont issus des projets CSTAR et Nespole!, et d'une évaluation des systèmes de dictée vocale proposée par l'AUPELF.

IV.1.a. *CStar120*

Le corpus *CStar120* a été enregistré lors de la préparation du démonstrateur du projet CSTAR pour la présentation de juillet 1999. Il est composé de 120 énoncés concernant la réservation touristique, cadre d'expérimentation du projet CSTAR. Ces énoncés sont extraits du corpus réservation touristique. Deux locuteurs de sexe masculin ont participé à son enregistrement, dont l'un n'était pas français de naissance. La description du corpus est donnée dans le tableau suivant :

Code locuteur	**Natif français**	**Nombre de phrases**	**Nombre de mots**
HB	oui	20	130
ZK	non	100	997
Total :		120	1127

Tableau VI.1 : Informations sur le corpus CStar120

Nous pouvons remarquer que le locuteur de langue maternelle française n'est pas celui qui a réalisé le plus d'enregistrements. De plus, nous devons signaler que l'autre locuteur a commis des erreurs de prononciation dans certaines phrases. Par exemple, dans "dois-je vous verser des arrhes ?", le mot *arrhes* a été prononcé "*arrhèsse*". Ces erreurs ont été laissées dans le corpus.

Tous les résultats de reconnaissance obtenus sur le corpus *CStar* l'ont été sur la base des règles de transcription suivantes. Les mots composés, comme *d'accord*, ont été considérés comme un seul mot par le système d'alignement. Il en va de même pour les nombres comme *dix-sept*, *dix-huit*, etc. De plus, les inversions des pronoms sujets dans les formes interrogatives l'ont aussi été. Ainsi, *dois-je* et *pouvez-vous*, sont aussi comptés comme un seul mot. Cette manière de dénombrer les mots a aussi été employée pour les comptes affichés dans le tableau précédent.

IV.1.b. Corpus *Nespole!-G711*

Ce corpus a été produit au moyen du système de démonstration du projet Nespole!. Les signaux ont été enregistrés après codage-décodage en G711 [REF] (codage acoustique 8000 Hz, 8 bits avec une quantification μ-law) et transmission avec perte sur un réseau. Cette perte est négligeable, mais non nulle, car la session d'enregistrement a été réalisée sur un réseau local.

Cette base contient au total 77 signaux, prononcés par un seul locuteur, correspondant à des tours de parole extraits eux-mêmes du corpus *Nespole!*. C'est pourquoi nous appellerons ce corpus *Nespole!-G711* pour le différencier du corpus textuel *Nespole!*. Le tableau suivant présente ses caractéristiques :

Code locuteur	**Natif français**	**Nombre de phrases**	**Nombre de mots**
LB	oui	77	921

Tableau VI.2 : Informations sur le corpus Nespole!-G711

L'intérêt principal de ce corpus est le fait qu'il est constitué de phrases longues, presque 12 mots en moyenne, soit 1/3 de plus que les 9 mots du corpus *CStar120*. Les signaux contiennent des tours de parole comportant plusieurs phrases successives, des hésitations, des répétitions, etc. Enfin, le vocabulaire de la tâche Nespole! est beaucoup plus étendu que celui de CSTAR puisqu'il compte, en tout cas pour le système de reconnaissance, plus de 20000 mots.

IV.1.c. Corpus de test de la campagne d'évaluation AUPELF

L'ARC (Action de Recherche Coordonnée) B1 de l'AUPELF [Web 11], consistait en l'évaluation comparative de systèmes de dictée vocale [Chibout et al. 99], [Dolmazon et al. 97]. La tâche de reconnaissance comprend 299 phrases extraites du journal « Le Monde », prononcées par 20 locuteurs, 10 hommes et 10 femmes. Nous appellerons ce corpus « Aupelf ». Le tableau ci-après indique les caractéristiques de ce corpus pour chacun des locuteurs. Le code du locuteur indique grâce à sa dernière lettre, si le locuteur est un homme ('m') ou une femme ('f').

Code locuteur	Natif français	Nombre de phrases	Nombre de mots
l02m	oui	21	554
l09f	oui	18	522
l06f	oui	20	670
l10m	oui	23	624
l17f	oui	13	384
l15m	oui	9	180
l13f	oui	13	262
l01m	oui	10	299
l20m	oui	18	612
l16m	oui	3	137
l18f	oui	13	400
l05f	oui	16	379
l12m	oui	10	376
l07f	oui	19	677
l04f	oui	18	528
l03m	oui	19	595
l19m	oui	20	605
l11f	oui	11	333
l14f	oui	13	446
l08m	oui	12	375
Total :		299	8958

Tableau VI.3 : Informations sur le corpus Aupelf

Ce corpus contient plus de données que le précédent. De plus, les règles de transcription qui s'y appliquent sont différentes. Ainsi, les mots composés, conformément aux recommandations de l'AUPELF, elles-mêmes provenant à l'origine du NIST, sont éclatés. Ainsi, *d'accord* est cette fois considéré comme deux mots. Si une erreur survient, elle engendre alors deux erreurs pour le système d'alignement : une substitution du mot plus une suppression. Pour les expérimentations impliquant cette base, nous utiliserons comme outil d'alignement *sclite* du NIST.

Cette base *Aupelf* étant *état de l'art*, de par le fait que plusieurs équipes de recherche ont évalué leur système de reconnaissance dans le cadre de l'ARC B1, nous avons aussi suivi ces recommandations. Nos résultats sont ainsi comparables à ceux des autres laboratoires. De plus, même si le contenu de ce corpus sonore est loin de notre objectif, le dialogue, nous nous devions de trouver un point de comparaison pour ne plus évaluer nos modèles de langage uniquement sur des données propriétaires.

IV.2. Corpus *CStar120*

Comme précédemment, nous utiliserons une tâche limitée pour faire nos expériences. Cette tâche sera de nouveau la tâche de réservation touristique de type CSTAR. Dans un premier temps, nous referons nos expériences sur l'influence de la taille du corpus d'apprentissage comme nous l'avons fait dans la partie précédente. Par la suite, nous étudierons, toujours dans les mêmes conditions, comment le réglage de l'extraction des blocs minimaux peut faire varier le taux de reconnaissance.

IV.2.a. Influence de la taille du corpus d'apprentissage

Reprenons les expériences menées dans la première partie de ce chapitre, cette fois-ci dans le cadre de la reconnaissance de la parole. Nous utiliserons le même découpage, en bloc de 2 millions de mots, du texte filtré sur *WebFr*, le vocabulaire de la tâche CSTAR et une taille de blocs minimaux de 3. Nous mesurons l'évolution du taux de reconnaissance sur le corpus *CStar120*. Le graphique ci-dessous montre les résultats obtenus :

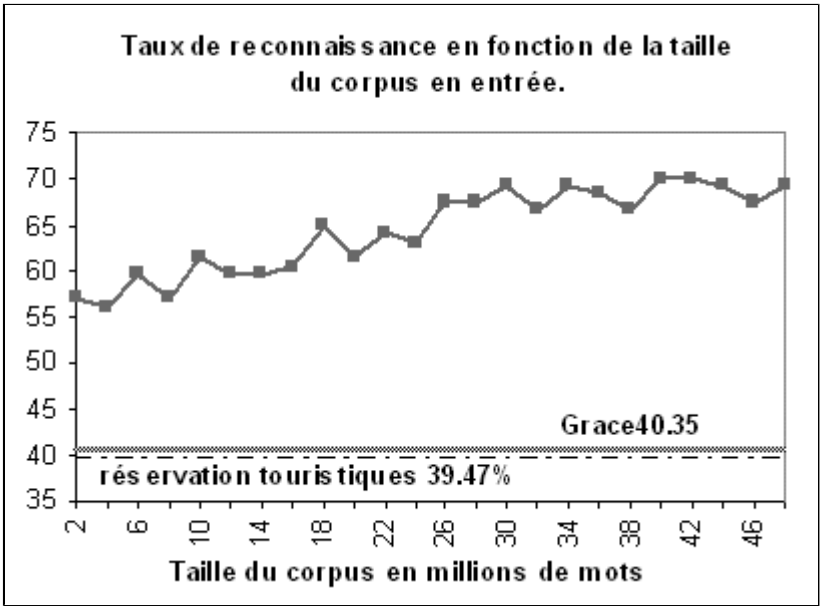


Figure VI.4 : évolution du taux de reconnaissance en fonction de la taille du corpus en entrée sur CStar120

D'abord, nous trouvons dans la partie basse de ce graphique deux références représentant les résultats obtenus en apprenant nos modèles de langage sur les corpus *réservations touristiques* et *Grace*. Nous voyons que notre hypothèse sur l'inadéquation de *Grace* se vérifie car nous obtenons seulement 40,35% de taux de reconnaissance. Ce qui est surprenant, c'est que le corpus *réservations touristiques* est moins bon, même si la différence n'est pas significative. Pourtant, on aurait naturellement penser que l'apprentissage d'un modèle de langage sur ce corpus aurait permis de très bien modéliser les énoncés de *CStar120*, qui rappelons-le sont extraits du même corpus. Mais il faut reconnaître que les efforts menés pour obtenir ce corpus dans l'espoir de sa représentativité de la tâche n'ont pas porté leurs fruits. Le problème dans ce cas, reste la taille très réduite de ce corpus, certainement insuffisante pour le calcul des probabilités de notre modèle de langage.

A contrario, *Grace*, de par sa plus grande taille représente mieux certaines structures du français, mais il reste lui aussi autour de 60% d'erreurs, car il ne présente pas les formes dont nous avons besoin pour le langage oral. Ces deux corpus se révèlent être de très mauvais représentants de la tâche de dialogue pour la réservation touristique.

Si nous nous intéressons maintenant aux modèles de langage trigrammes appris sur des documents de la Toile, nous pouvons constater que dans le pire des cas, le taux de reconnaissance est de 15% supérieur aux deux corpus de références (pour 4 millions de mots en apprentissage). De plus, la progression, même si elle est irrégulière, présente une tendance linéaire très nette. À la fin, pour un corpus de 46 millions de mots, nous obtenons un taux de reconnaissance de 74% ce qui représente une réduction du taux d'erreur de 56%. Si l'on prend maintenant la totalité du corpus textuel extrait de *WebFr*, c'est-à-dire environ 145 millions de mots, le taux de reconnaissance obtenu par le système est de 78%.

Nous pouvons comparer ces résultats avec ceux de perplexité obtenus précédemment. En effet, il est aisé de constater que malgré sa faible variation après le passage à un apprentissage sur plus de 28 millions de mots, le taux de reconnaissance ne cesse d'augmenter. De plus, si nous calculons la perplexité uniquement sur les phrases à reconnaître (1127 mots), nous obtenons quasiment la même courbe de perplexité que sur tout le corpus *réservations touristiques*. Nos conclusions restent, dans ce cas, les mêmes que dans [Vaufreydaz et al. 99c]. La perplexité n'est donc pas une bonne mesure pour juger de la pertinence d'un modèle de langage pour la reconnaissance de la parole.

IV.2.b. Influence des caractéristiques des blocs minimaux

À ce stade, nous pouvons nous demander comment améliorer encore les résultats. Nous avons supposé qu'on peut y arriver en "choisissant mieux" les blocs minimaux. Il nous faut donc étudier en détails les paramètres régissant la production du corpus d'apprentissage par la méthode des blocs minimaux. Le premier est la taille minimale des blocs que nous gardons. Le second, booléen, indique si l'on ne souhaite garder que les phrases complètes. Dans le précédent chapitre, nous avons déjà détaillé l'effet de ce changement sur la quantité de corpus que nous générions.

IV.2.b.1. Étude de la taille des blocs minimaux

La taille des blocs minimaux influe non seulement sur la taille des corpus d'apprentissage, mais aussi sur la structure des données obtenues. En effet, si l'on regarde quels sont les événements, à savoir unigrammes, bigrammes et trigrammes, qu'il est possible de recenser en considérant les différentes tailles de blocs minimaux, on remarque des différences. Or, nous avons déjà indiqué que, pour ne pas avoir à employer des méthodes de repli ou de recombinaison, il fallait maximiser la couverture en trigrammes du modèle.

Nous allons donc maintenant étudier l'apprentissage possible en fonction de la taille des blocs. Nous rappelons, comme nous l'avons dit dans le chapitre précédent, que lors du passage d'une longueur L à une longueur L+1, la taille du corpus se réduit d'environ 50%. Cela signifie que, en moyenne, la moitié d'un corpus généré avec une taille de bloc L est constituée de blocs de taille L. Les remarques que nous allons faire sur des corpus avec des

tailles de blocs fixées sont alors valables sur 50% de la masse d'apprentissage disponible. Il est alors clair que cette partie de l'apprentissage a une très grosse influence sur le modèle final. L'autre moitié du corpus d'apprentissage est alors concernée par les remarques sur les corpus extraits avec L+1 comme paramètre.

La première évidence en fonction de nos objectifs, c'est-à-dire des modèles trigrammes, est que la valeur inférieure de taille minimale des blocs est 3. Avec une valeur plus faible, nous calculerions des probabilités d'apparition de bigrammes et d'unigrammes qui ne seraient pas directement en relation avec la tâche considérée, puisqu'ils ne feraient pas forcement parti de contextes trigrammes basés sur le vocabulaire. Nous risquerions d'apprendre des contextes qui sont hors de la tâche visée.

Nous allons maintenant détailler quels sont les événements que nous allons trouver dans notre corpus d'apprentissage sur des blocs de taille 3, 4 et 5 :

- cas de L=3

Sur *WebFr*, le corpus d'apprentissage sur la tâche de réservation touristique comporte 145 millions de mots. La figure VI.5 présente les événements que l'on dénombre dans un bloc de taille 3. Nous trouvons en trait plein les unigrammes, en tirets courts les bigrammes et en tirets longs les trigrammes.

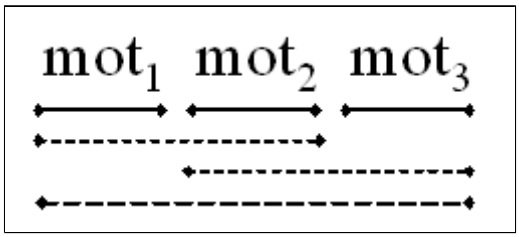


Figure VI.5 : apprentissage sur des blocs minimaux de longueur 3

Sur un tel bloc, nous voyons qu'il est possible de compter 3 unigrammes, 2 bigrammes et 1 trigramme seulement. En fait, sans méthode de repli, nous avons seulement appris comment prédire mot_3 en sachant mot_1 et mot_2. Si l'on considère qu'il est statistiquement tout à fait possible de ne trouver mot_3 qu'en dernière position dans des blocs de taille 3, alors nous ne possédons aucune information sur la capacité de mot_3 à prédire ses successeurs.

Dans cet exemple, l'information contextuelle d'un bloc minimal de taille 3 pour un modèle trigramme est $IC_3(bloc) = 0$. Nous n'avons appris aucune information contextuelle complète pour aucun mot. Dans l'éventualité d'employer un modèle bigramme, cette valeur serait $IC_2(bloc) = 1$. Pourtant, cela ne représente en fait que 1/3 de la masse d'apprentissage, ce qui est assez peu.

- cas de L=4

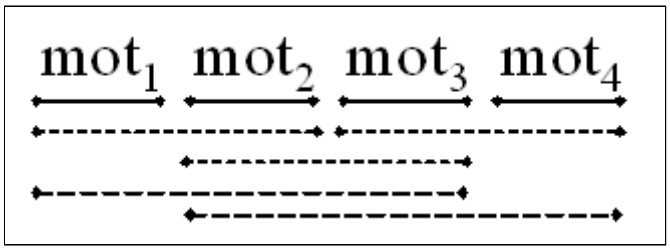


Figure VI.6 : apprentissage sur des blocs minimaux de longueur 4

Dans cette configuration, nous trouvons 2 trigrammes, 3 bigrammes et 4 unigrammes. Comme précédemment, l'information contextuelle dans ce bloc minimal, en contexte trigramme est nulle. Elle devient, pour un contexte bigramme, $IC_2(bloc) = 2$ ce qui nous donne 50% d'information critique dans ce bloc.

- cas de L=5

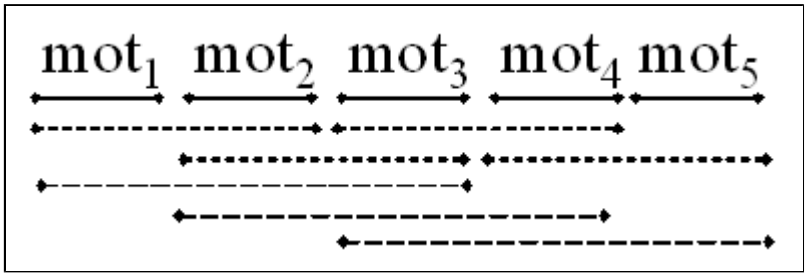


Figure VI.7 : apprentissage sur des blocs minimaux de longueur 5

D'après la figure VI.7, un bloc de taille 5 contient 5 unigrammes, 4 bigrammes et 3 trigrammes. L'information contextuelle $IC_3(bloc)$ vaut 1 et est non nulle pour la première fois. Avec un bloc minimal de taille 5, nous nous trouvons pour la première fois dans la position où un mot a fourni une information contextuelle complète. D'après notre métrique, nous pouvons penser que la taille inférieure des blocs que nous devions générer, pour apprendre des modèles trigrammes, est de 5. Cela revient en fait à dire qu'avec ce bloc, nous avons modélisé complètement le mot_3.

IV.2.b.2. Variation du taux de reconnaissance

La validation du résultat précédent, en reconnaissance de la parole cette fois, se retrouve dans le tableau VI.4.

Taille des blocs minimaux	Taille du corpus en mots	IC_2	IC_3	Taux de reconnaissance
3	145 millions	1	0	78%
4	71,5 millions	2	0	80%
5	45 millions	3	1	88%

Tableau VI.4 : variation du taux de reconnaissance en fonction de la taille des blocs minimaux

Ces taux de reconnaissance valident nos hypothèses et notre mesure, l'information contextuelle, pour la reconnaissance de la parole. En effet, on voit bien que lorsque la valeur d'information contextuelle, augmente au niveau bigramme, le taux de reconnaissance augmente également. De plus, lorsque cette valeur, pour une valeur de contexte de 3, passe à une valeur non nulle, le taux de reconnaissance atteint sa valeur maximale. En effet, 88% est le maximum expérimental qu'il est possible d'obtenir sans rajouter les mots hors vocabulaire présents dans les signaux à reconnaître. C'est pour cette raison que nous n'avons pas poussé notre expérimentation en prenant des blocs de taille minimale égale à 6.

IV.2.b.3. Phrases complètes

Au vu de la taille des corpus obtenus lorsque nous réglons notre filtre sur des phrases complètes (quelques dizaines de milliers de mots), nous ne pouvons pas espérer avoir de bons résultats. Si l'on prend, comme nous l'avons vu dans la section précédente, une taille de blocs minimaux égale à 5, nous obtenons un corpus de phrases complètes de 46500 mots. Le tableau ci-après récapitule les résultats obtenus avec différentes sources de corpus de taille similaire.

Corpus	Nombre de mots	Taux de reconnaissance
réservations touristiques	97253	39,47%
Nespole!	42598	32,85%
WebFr filtré	46500	36%

Tableau VI.5 : comparaison du taux de reconnaissance de WebFr, filtré pour n'obtenir que des phrases complètes, avec les corpus réservations touristiques et Nespole!

Même si l'avantage de posséder des phrases complètes pour l'apprentissage de nos modèles peut se révéler un atout, il est évident qu'une fois encore, dans ce cas, la taille plus que réduite du corpus d'apprentissage ne nous apporte pas assez d'information pour nos

modèles de langage. Même en utilisant *WebFr4*, sur une tâche aussi limitée que la réservation touristique, nous obtenons un petit corpus d'environ 123000 mots qui reste insuffisant pour nos apprentissages.

L'utilisation de ce paramètre ne peut porter ses fruits que dans l'éventualité de posséder un nombre plus conséquent de mots différents pour maximiser l'obtention de phrases complètes. De plus, la méthode des blocs minimaux ayant été optimisée pour le calcul de modèles de langage, et ces derniers fonctionnant à partir de contextes courts, il n'est alors pas critique de posséder des phrases complètes. De plus, cette option réduit fortement la taille des corpus, ceux-ci retrouvant une taille de l'ordre des corpus que nous possédions déjà. Elle ne sera donc pas utile pour l'apprentissage de modèles de langage. Nous verrons que cette option est cependant précieuse pour d'autres utilisations de nos outils comme pour la constitution de corpus de phrases pour l'enregistrement de signaux de parole [Vaufreydaz et al. 00]. Cela fera l'objet d'un prochain chapitre.

IV.2.c. Conclusion

Au vu des résultats que nous venons d'exposer, nous pouvons dire que plusieurs paramètres doivent êtres choisis avec soin. Premièrement, la taille des blocs minimaux n'influe pas seulement sur la taille du corpus. Même en produisant un corpus contenant 100 millions de mots de moins, en augmentant la taille minimale des blocs de 3 à 5, le gain en taux de reconnaissance est de 10%, ce qui représente une réduction du taux d'erreur de 54%. Ensuite, l'utilisation de phrases complètes n'apporte pas de très bons résultats, même si, à taille de corpus équivalente entre celui obtenu à partir de *WebFr* et *Nespole!*, le premier donne un résultat significativement supérieur au second. Encore une fois, Internet est une source de texte, même avec ce paramétrage extrême, de qualité supérieure à des collectes de données même en conditions réelles.

IV.3. Corpus *Nespole!-G711*

Le modèle de langage utilisé pour la reconnaissance de ce corpus est basé sur un vocabulaire de 20524 mots. Il a été obtenu entièrement automatiquement en utilisant la chaîne complète de construction de corpus présentée dans le chapitre précédent et dans [Vaufreydaz et al. 01a]. Premièrement, le vocabulaire de base de l'application visée, la réservation touristique étendue du projet Nespole!, a été défini en utilisant les lexèmes présents dans les corpus *réservations touristiques* et *Nespole!*. De plus, les quelques mots ajoutés empiriquement pendant la préparation du démonstrateur du projet CSTAR ont enrichi ce vocabulaire pour atteindre un peu plus de 2500 mots. La seconde phase a consisté à compléter ce vocabulaire par des vocables plus généraux pour réduire la probabilité d'apparition de mots hors vocabulaire lors de la phase de reconnaissance. Cela a été réalisé en augmentant le vocabulaire avec les graphèmes français les plus présents dans *WebFr4*. À ce stade, le vocabulaire fait exactement 20000 mots, une valeur raisonnable que nous avons fixée. Nous avons ensuite ajouté les mots composés, formés de mots courts du vocabulaire les plus présents sur *WebFr4*, et atteint finalement les 20524 mots. Le dictionnaire phonétique final de RAPHAEL comporte 27117 entrées.

Le modèle de langage a été appris sur *WebFr4* filtré par la méthode des blocs minimaux avec une taille de bloc égale à 5. Le corpus d'apprentissage comporte 1587142200 mots, ce qui confirme notre hypothèse sur la quantité d'information qu'il est possible d'extraire de *WebFr4*, comme nous l'avons vu dans l'étude d'Internet. Le modèle de langage comporte finalement 1960813 bigrammes et 6413376 trigrammes. Le taux de reconnaissance que nous obtenons sur ce corpus est de 79,1%. De par la complexité des énoncés contenus dans les signaux à reconnaître et l'aspect très général du vocabulaire, la définition du vocabulaire de base n'ayant d'utilité que de garantir la présence de ces mots dans le système, ce résultat est satisfaisant. Si l'on considère qu'aucune adaptation particulière n'a été réalisée pour augmenter ce résultat, obtenu de façon entièrement automatique, et que, de plus, ce taux de reconnaissance nous permet de réaliser la phase suivante dans le système Nespole!, à savoir la traduction, nous considérerons ce résultat comme répondant à nos objectifs initiaux [Vaufreydaz et al. 01a]. Nous travaillons cependant à son amélioration au moment de la rédaction de ce manuscrit.

IV.4. Corpus *Aupelf* de l'évaluation de l'ARC B1 de l'AUPELF

IV.4.a. Données mises à la disposition des participants

Pour permettre l'évaluation comparative des systèmes de reconnaissance, il est nécessaire de fournir à tous la même base de travail. Les participants ont donc pu profiter de ressources communes, sous forme de corpus sonores et textuels, pour la mise au point de leurs outils. Cette catégorie d'évaluation avec des données imposées se nomme *P0*. Le lecteur trouvera toutes les informations sur l'ARC B1 de l'Aupelf sur le site [Web 13].

Les corpus sonores fournis étaient *BREF-80* et *BREF-TOTAL* [Lamel et al. 01] avec toutes les transcriptions correspondantes. Pour l'apprentissage des modèles de langage, les textes du journal « *Le Monde* » des années 1987 et 1988. Enfin, le vocabulaire des systèmes était lui aussi donné sous la forme d'une liste de 20000 mots. Cette liste correspond aux mots les plus fréquents dans les années 1987 et 1988 du journal.

Le corpus de test comprend 299 phrases correspondant à des extraits lus de « *Le Monde* » qui ne se retrouvent pas dans les données d'apprentissage des modèles de langage. Elles ont été choisies pour minimiser le taux de mots hors vocabulaire par rapport à la liste de 20000 mots fournis.

IV.4.b. Résultats des participants de l'évaluation

Les équipes de recherche ayant participé à cette évaluation sont l'INRS-Télécommunication et le CRIM de Montréal, le LIMSI à Orsay et le LORIA à Nancy. Les résultats ainsi que les divers paramètres des systèmes sont résumés dans le tableau suivant :

Équipe de recherche	Vocabulaire augmenté	Apprentissage augmenté	Taux d'erreur
INRS	non	non	38,49%
CRIM	non	non	38,78%
LIMSI	oui	oui	11%
LORIA	non	non	31,85%

Tableau VI.6 : taux d'erreur des systèmes ayant participé à l'évaluation

Ces résultats sont ceux présentés dans [Savariaux et al. 97], [Lazaridès et al. 98] et [Zitouni 00]. Comme nous pouvons le voir, le meilleur résultat est de 89% de reconnaissance. Pourtant, comme le souligne [Zitouni 00] page 140, le système ayant obtenu ce résultat a été entraîné sur 270 millions de mots (contre 40 pour les autres) et réalise de plus une adaptation acoustique aux phrases à reconnaître. De plus, le vocabulaire du système est supérieur aux 20000 mots fournis, ce qui diminue les problèmes dus aux mots hors vocabulaire. Les autres systèmes ont tous des taux d'erreur supérieurs à 30%. Le meilleur système de ce second lot est celui du LORIA. Viennent ensuite les systèmes du CRIM et de l'INRS. Il est à noter que des résultats meilleurs ont été publiés par ces équipes de recherche en utilisant diverses techniques d'adaptations de leur système. Ceci n'est toutefois pas le propos de notre comparaison puisque nous n'utilisons aucune de ces techniques.

IV.4.c. Résultat avec la méthode des blocs minimaux

Nous allons maintenant voir quels sont les résultats que nous pouvons obtenir en utilisant Internet comme source de texte pour cette tâche. Le calcul de notre modèle de langage est entièrement automatique. La seule modification que nous ayons apportée à notre méthode concerne les mots composés. Nous n'avons en effet pas ajouté de mot composé au vocabulaire de base pour respecter le vocabulaire défini pour l'évaluation. Les blocs minimaux utilisés, conformément à ce que nous avons vu au début de ce chapitre sont de longueur minimale égale à 5. Le corpus d'apprentissage obtenu sur *WebFr4* fait 1292328777 mots. Le modèle de langage résultant comprend 2305645 bigrammes et 7478136 trigrammes. Le modèle acoustique est, quant à lui, entraîné sur *BREF-80*, ce qui représente environ 10 heures de parole. Les autres participants ont eux entraîné leurs systèmes sur *BREF-TOTAL* qui représente plus d'une centaine d'heures de signal.

Le taux de reconnaissance que nous obtenons pour cette tâche, avec un treillis de mots et sans aucune modification manuelle des hypothèses produites par RAPHAEL pour traiter les homophones par exemple, est de 63%, soit 37% d'erreurs [Vaufreydaz et al. 01b]. Nous voyons que ce résultat totalement automatique est du même ordre que la plupart des systèmes ayant participé à l'évaluation. De plus, dans ce cas, Internet se montre aussi satisfaisant comme source de texte que le texte fourni pour l'évaluation, sachant que les journaux nationaux en ligne n'autorisent pas les robots à collecter leurs documents. Les phrases à reconnaître étant des énoncés tirés du journal « *Le Monde* » ainsi que le corpus

d'apprentissage donné aux participants, ce sont donc des textes jumeaux. L'intérêt d'Internet pour la modélisation du langage, même de type journalistique, est validé par ces résultats.

Nous sommes actuellement en train de travailler sur les modèles acoustiques de RAPHAEL. Nous allons bientôt disposer d'un modèle entraîné lui aussi sur *BREF-TOTAL*, ce qui devrait nous permettre d'être dans des conditions plus proches de celles des participants à l'évaluation. Nous devrions avoir un gain non négligeable en augmentant la taille du corpus d'apprentissage acoustique. Nous menons en parallèle, en collaboration avec le laboratoire LIA d'Avignon, une étude permettant de déterminer la contribution relative de l'acoustique et du modèle de langage dans le taux d'erreur que nous obtenons. Pour cela, nous sommes en train de réaliser des tests croisés entre les systèmes acoustiques et les modèles de langage. En comparant les taux d'erreur de notre modèle de langage avec le système acoustique du LIA et le modèle de langage du LIA avec notre système acoustique, nous pourrons déterminer quels sont le ou les points à améliorer. Nous menons ces tests parallèlement à la rédaction de ce manuscrit.

IV.4.d. Influence de l'évolution d'Internet

Les signaux de la campagne d'évaluation AUPELF représentant la tâche la plus difficile pour Raphaël, nous avons décidé d'étudier la différence entre un apprentissage sur *WebFr* et *WebFr4* pour nos modèles de langage. Les paramètres sont dans les deux cas les mêmes que précédemment. La seule différence réside dans le fait que, lors de l'emploi de *WebFr*, le module de correction orthographique est activé.

En utilisant *WebFr*, le taux de reconnaissance que nous obtenons est de 51% (49% d'erreurs). Si nous le comparons avec les 37,2% d'erreurs pour *WebFr4*, nous voyons une réduction de 30% du taux d'erreur. Cela valide nos postulats sur l'intérêt grandissant d'Internet pour la modélisation du langage. Un premier facteur intervenant dans ce gain est l'accroissement incessant du nombre de documents que l'on peut trouver sur la Toile. Le second concerne l'hétérogénité de ces documents et l'augmentation du nombre de vocables qu'ils contiennent. Ces deux facteurs concourent simultanément à l'amélioration de la qualité de nos modèles de langage. Ce résultat confirme ce que nous avions déjà énoncé dans le chapitre « *Étude d'Internet* », où nous nous étions fondé uniquement sur la représentation des pronoms personnels, sur le nombre de mots différents et aussi sur la taille croissante des corpus.

Conclusion

Les résultats présentés dans ce chapitre montrent bien l'intérêt, du point de vue de la reconnaissance de la parole, de la méthode des blocs minimaux. L'étude statistique de la représentation de l'information au sein d'un corpus extrait de la Toile a montré une répartition uniforme de celle-ci, très certainement due à la méthode de collecte aléatoire de notre robot Clips-Index. Pourtant, chaque sous-partie apporte une nouvelle variété de construction qui semble intéressante. Nous pouvons maintenant affirmer qu'il est assez rapide de trouver toutes les constructions de base du langage, même oral, sur Internet. L'ajout de quantité ne

fait qu'accroître la variété et affiner les probabilités du modèle de langage. Cette remarque fait écho à celle de [Antoine et al. 01], que nous avons déjà citée dans le chapitre I selon laquelle, pour les dialogues, il y aurait un noyau minimal constant, augmenté par des constructions spécifiques à la tâche visée. Nous avons aussi montré, comme d'autres, que la perplexité, même si elle présente des propriétés intéressantes, est assez mal corrélée avec le taux de reconnaissance final.

En ce qui concerne la tâche de reconnaissance, les modèles obtenus par la méthode des blocs minimaux apportent de bons résultats en terme de taux de reconnaissance. Cette méthode, employée sur des corpus issus de la Toile, pallie le manque de données pour l'apprentissage de modèles statistiques. De plus, en rajoutant des filtres avant le calcul final du modèle de langage, il est tout à fait envisageable de construire d'autres types de modèles comme des modèles n-classes. Notre démarche semble donc ouvrir une voie nouvelle et efficace pour l'obtention de corpus intéressants dans le cadre de la modélisation statistique du langage.

Nous arrivons ainsi à la fin de la seconde partie de cette thèse qui présentait Internet et son utilisation dans la modélisation du langage pour la reconnaissance de la parole spontanée. Nous allons maintenant aborder d'autres utilisations que nous pouvons faire d'Internet dans le cadre d'applications liées à la reconnaissance de la parole.

Partie III :
Autres travaux

Chapitre VII : Vers une détection de thème utilisant les *newsgroups*

Chapitre VII : Vers une détection de thème utilisant les *newsgroups*

Présentation du chapitre

Les résultats obtenus précédemment montrent un grand progrès dans l'apprentissage de modèles de langage pour un contexte donné. Comme en traduction assistée par ordinateur (TAO), il semble impossible de construire un système de reconnaissance de la parole de haute qualité pour tous les contextes. La solution souvent retenue est de développer un modèle de langage pour chaque contexte et de passer de l'un à l'autre grâce à un module de détection de thème, qui détermine le thème courant du discours. Cette identification de thème est un problème majeur et pour le résoudre nous avons naturellement cherché à utiliser de nouveau Internet. Cependant, notre approche diffère sur plusieurs points. En place de l'ensemble de modèles adaptés aux thèmes, nous utiliserons un modèle de langage générique très grand vocabulaire que nous pondérerons en fonction du thème courant. De plus, visant des applications en dialogue, il nous faut détecter le thème au cours d'un seul énoncé.

Nous présentons donc dans ce chapitre les premiers travaux que nous avons menés sur cette détection de thème. Notre classification thématique est basée sur une arborescence de *newsgroups*. Nous verrons comment nous formulons cette détection de thème au cours d'un seul énoncé et comment elle est intégrée au sein de notre système de reconnaissance. Nous présenterons enfin les premiers résultats que nous avons obtenus.

I. Introduction

I.1. Pourquoi détecter le thème ?

La détection du thème, d'un discours ou d'un texte, dans une tâche de modélisation du langage se base sur le fait que certains termes sont plus ou moins probables si l'on parle d'un thème ou d'un autre. De plus, comme nous l'avons indiqué dans le chapitre V, section I, certains mots peuvent être employés différemment, c'est-à-dire dans d'autres contextes, selon le thème. C'est pourquoi il peut être utile de trouver le thème du discours en cours pour pouvoir changer, à la volée, les pondérations des mots au sein du modèle de langage et ainsi mieux refléter le discours. Cependant, dans le cadre d'un dialogue, le changement de thème peut intervenir sur un ou deux tours de parole seulement. Dans ce cas-là, il est très difficile de trouver le thème en cours et de le prendre en compte au sein du modèle de langage. Dans ce chapitre, nous tenterons d'apporter un embryon de réponse à ce problème de détection de thème au cours d'un seul énoncé.

I.2. Utilisation des *newsgroups*

Comme nous l'avons déjà indiqué, les *newsgroups* sont organisés en thèmes et en sous-thèmes de manière hiérarchique. Ainsi, si l'on ne s'intéresse qu'aux newsgroups de langue française, il est suffisant de regarder la hiérarchie 'fr.' qui contient la liste des groupes en français. Ensuite, la branche 'fr.comp' contient les groupes parlant d'ordinateurs. Effectivement, 'comp' vient de *computer* car le nom cette branche vient de la branche anglophone 'comp.', plus ancienne, contenant des groupes parlant d'ordinateurs. En dessous, nous trouvons, par exemple, 'fr.comp.ia' et 'fr.comp.mail' qui traitent respectivement d'intelligence artificielle et de messagerie électronique. Nous pouvons voir que chaque fois que l'on descend dans l'arborescence des groupes, on obtient une thématique plus raffinée. C'est cette propriété de thématique arborescente que nous nous proposons d'utiliser pour tenter de modéliser le changement de thème au cours d'un seul énoncé.

II. Approches existantes

A notre connaissance, il y a 3 principales approches de modélisation du langage qui intègrent une connaissance thématique. Il s'agit des modèles cache et *trigger*, des modèles à *mixture* et des modèles intégrant la probabilité du thème.

II.1. Modèles cache et *trigger*

Les modèles cache et *trigger* peuvent permettre de prendre en compte les mots précédents pour tenter de détecter le thème. [Rosenfeld 94] a utilisé le modèle *trigger* en le combinant avec d'autres sources comme des n-grammes. Ce modèle sert alors à insérer des contraintes dans le modèle généraliste.

[Bigi 00] a employé le modèle cache pour réaliser la détection de thème. Même avec un cache réduit (5 mots), elle a ainsi obtenu un gain de perplexité. Elle utilise une méthode

d'interpolation et dans ce cas, il faut non seulement optimiser la gestion du cache mais aussi les facteurs d'interpolation. Cela s'avéra impossible pour le thème *Histoire* par exemple ([Bigi 00] page 64), ce qui montre la difficulté de l'estimation des paramètres d'interpolation.

II.2. Modèles à *mixture*

Les modèles à *mixture* [Kneser et al. 93a] généralisent le principe d'interpolation. Ils sont nommés ainsi car ils sont proches des *mixtures* utilisées en modélisation acoustique. À partir d'un ensemble, possiblement très grand, de modèles de langage pour chacun des thèmes possibles, le but est d'obtenir le score final en combinant les *N* modèles thématiques comme cela est indiqué dans l'équation suivante :

$$P(m_i \mid h) = \sum_{k=1}^{N} \lambda_k . P_k(m_i \mid h)$$

Équation VII.1 : formule d'interpolation d'un modèle à mixture

Dans l'équation VII.1, les facteurs λ_k sont les facteurs d'interpolation des modèles thématiques. Il est difficile d'obtenir ces facteurs, sinon empiriquement ou en les optimisant sur un corpus de test. De plus, le lissage des probabilités pose problème même si il est possible, en fixant une fois pour toutes les λ_k, de construire un modèle en utilisant les comptes pondérés de chacun des sous-modèles thématiques comme l'explique [Bellagarda 01a].

II.3. Modèles intégrant la probabilité du thème

Une autre approche consiste à utiliser directement la probabilité d'apparition d'un mot dans un thème et de ce même thème sachant un historique donné [Gildea et al. 99]. L'équation VII.1 se transforme, en notant t_k le $k^{ième}$ thème, pour devenir :

$$P(m_i \mid h) = \sum_{k=1}^{N} P(m_i \mid t_k) . P(t_k \mid h)$$

Équation VII.2 : formule d'interpolation d'un modèle à base de thèmes

Cette approche nous paraît plus naturelle. Elle exprime la probabilité d'un mot dans un thème et de ce thème sachant l'historique. En cela, elle se rapproche d'un modèle cache même si elle correspond à l'interpolation de plusieurs thèmes. Notre proposition sera basée sur une méthode de formulation proche de celle-ci, c'est-à-dire incorporant la probabilité d'un mot dans un thème et un conditionnement du résultat en fonction du thème ainsi trouvé.

III. Proposition

Notre proposition part d'une constatation que nous avons déjà mentionnée plusieurs fois dans ce manuscrit. L'utilisation d'un mot, en fonction du thème ou de la tâche, peut se faire dans différents contextes. Notre idée est alors proche de celle de [Rosenfeld 94] avec les

modèles *trigger* : utiliser le thème pour contraindre un modèle généraliste très grand vocabulaire. Nous sommes dans le cas contraire des systèmes à base de *mixture*. Cependant, nous voyons des contraintes plus longues que le simple mot. En effet, dans un thème donné, l'utilisation des trigrammes, donc des contextes d'un mot, peut varier par rapport au modèle généraliste.

Nous allons donc nous attacher, en fonction des mots contenus dans le contexte, à pondérer les probabilités des unigrammes, bigrammes et trigrammes de notre modèle généraliste. Pour cela, nous définissons des classes d'événements en fonction du thème courant et du type d'événement.

Pour les unigrammes, il existe deux classes : celle des mots présents dans le thème et celle des mots qui ne sont pas partie intégrante du thème. Pour les bigrammes, nous dénombrons trois classes. La première correspond aux bigrammes extérieurs au thème, c'est-à-dire composés de mots qui ne sont pas dans le thème considéré. La seconde est celle dont l'un des mots est contenu dans le thème. La dernière fait référence aux bigrammes dont les deux mots sont présents dans le thème courant. Selon le même raisonnement, il y a 4 classes de trigrammes.

Notre algorithme se propose alors, en fonction du thème courant (nous verrons plus loin comment nous le découvrons automatiquement), de calculer pour le modèle de langage généraliste, la pondération d'une classe d'événements dans le thème courant *t* selon la fonction suivante.

$$\mathrm{Pond}_t(\ classe\ ) = 1 - \frac{\text{Nombre d'éléments de la classe dans le thème}}{\text{Nombre d'éléments de même type dans le thème}}$$

Équation VII.3 : calcul de la pondération d'une classe d'événements pour un thème donné

Comme nous pouvons le voir, plus une classe d'événements est probable dans un thème, plus sa pondération baisse et *vise versa*. Cela permet, dans les thèmes très précis, donc contenant peu de mots, de fortement favoriser les classes ne contenant que des mots du thème car celles-ci sont peu nombreuses dans le modèle généraliste. *A contrario*, plus le thème est généraliste, donc plus il contient de mots, plus on donne de chance à des évènements hors thème d'apparaître, car ils sont moins nombreux. Cela peut paraître étrange de prime abord. Pourtant, la détection de thème n'étant pas parfaite, il peut s'avérer important de laisser une part de la recherche explorer les mots d'autres thèmes. Cela revient à prendre en compte les autres thèmes comme le font les approches à *mixtures*. L'avantage de cette technique réside dans le fait que, plus le thème courant est fin, moins on laisse de chance aux classes d'événements ne contenant que des mots hors thème d'apparaître.

La fonction de calcul de la probabilité d'un mot dans le modèle généraliste *M* en sachant son historique et le thème courant *t* est :

$$P_t(m_i \mid h) = P_M(m_i \mid h) \,.\, Pond_t(classe(h, m_i))$$

Équation VII.4 : calcul de la pondération d'un mot en fonction de son historique pour un thème donné

A tout moment, il est possible de calculer cette probabilité. Pourtant, lorsqu'aucun thème n'est encore déterminé, nous avons simplement la valeur de probabilité exprimée dans le modèle généraliste.

IV. Implémentation

Nous avons implémenté notre méthode de détection de thème et modifié les algorithmes de recherche que nous utilisons dans RAPHAEL pour prendre en compte la détection de thème lors de la reconnaissance. Le module de détection de thème repose sur l'arborescence des *newsgroups* et la modification de RAPHAEL consiste essentiellement à intégrer la détection de thème dans l'algorithme *tree-forward* avec plusieurs stratégies. Nous terminerons cette section par des évaluations de cette nouvelle méthode.

IV.1. Détection du thème

IV.1.a. Construction de l'arbre des thèmes avec les *newsgroups*

L'utilisation des *newsgroups* nous donne déjà une classification manuelle, puisque réalisée par les utilisateurs envoyant des messages, des documents en thèmes et sous-thèmes. Nous avons alors une représentation arborescente de thèmes. La figure VII.1 nous montre un extrait d'une branche de l'arborescence française.

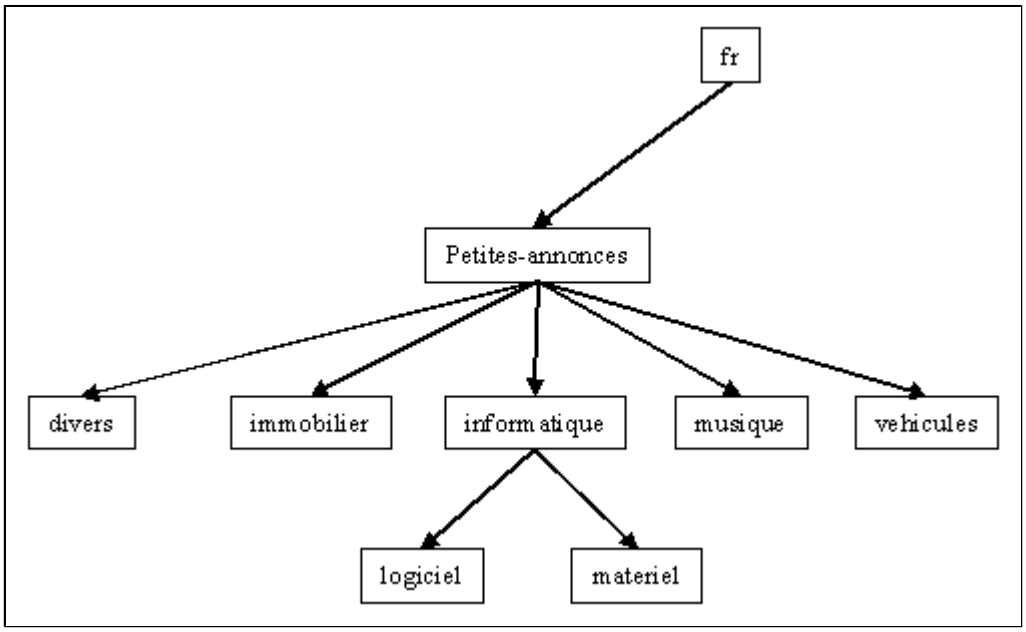


Figure VII.1 : extrait de l'arborescence de la hiérarchie 'fr.' des newsgroups

Sur cet exemple, nous voyons plusieurs feuilles et des nœuds intermédiaires. Les feuilles représentent les groupes comme 'divers' pour le groupe 'fr.petites-annonces.divers'. Par contre, la grande majorité des nœuds, comme ici 'petites-annonces', ne représente pas de groupe réel mais juste un découpage logique en thèmes plus fin. Nous allons devoir, dans ce cas, construire un pseudo-thème par concaténation des sous thèmes exposés. Cela permet d'obtenir, pour chaque nœud et chaque feuille, un thème qui, en fonction de sa profondeur dans l'arbre, est plus ou moins spécifique. Par ce mécanisme, nous obtenons aussi un thème, que nous appellerons le monde, qui n'est autre que le nœud 'fr' et qui regroupe tous les thèmes.

Si l'on considère le corpus de messages en provenance de *newsgroups NewFr*, Il y a 179 feuilles représentant les groupes de discussion, et 56 nœuds intermédiaires construits par concaténations de sous-thèmes. Ont été retirées de ce décompte, les branches de l'arbre dont le vocabulaire du nœud ou de la feuille était inférieur à 2000 vocables. En effet, certains groupes ne sont présents que pour des annonces ponctuelles sur la création d'un nouveau groupe ou pour faire de simples tests. On se retrouve donc avec des thèmes dont la masse d'apprentissage et le vocabulaire sont trop réduits.

IV.1.b. Formule de détection

Nous avons besoin d'une fonction de pondération rapide à mettre en œuvre, de façon à ne pas pénaliser les algorithmes de recherche. Comme nous l'avons dit, le but du module de modélisation du langage est de limiter l'espace de recherche pour ne pas avoir à explorer l'espace de toutes les solutions possibles. Pourtant, si ce module est trop lent, il perd l'une de ces fonctions principales. Nous avons, dans un premier temps, pensé à simplement construire un modèle unigramme pour chaque nœud et feuille de notre arbre. Pourtant les résultats furent médiocres. Le problème principal venait des mots ne portant pas forcément d'information thématique comme les pronoms 'le', 'la', 'les', etc. Il aurait été possible de ne pas les considérer dans notre calcul de thème. Cependant, cela va à l'encontre de la prise en compte de mots en contexte en fonction du thème, ces pronoms étant partie intégrante de la variété des constructions de mots possibles en fonction du thème. Nous nous sommes alors inspiré de la méthode TF-IDF (*Term Frequency - Inverted Document Frequency*) utilisée en Recherche d'Information [Salton et al. 83]. L'idée est toujours d'utiliser un modèle unigramme par nœud de l'arbre, mais cette fois en normalisant le résultat par rapport aux scores des mots dans le monde, c'est-à-dire notre pseudo-groupe 'fr'. Ainsi, l'importance relative d'un terme comme 'le' est supprimée car il est aussi très probable dans le monde. À l'opposé, un mot peu probable dans le thème et dans le monde, peut être relativement important dans le thème. La formule de calcul de la probabilité d'une séquence *S* de *i* mots s'exprime alors sous la forme :

$$P(S) = \prod_{i=1}^{n} \frac{P_{\text{thème}}(m_i)}{P_{\text{monde}}(m_i)}$$

Équation VII.5 : calcul de la probabilité d'une séquence de mots S pour un thème donné

L'avantage de cette formule est d'être très facilement calculable sous la forme d'une addition car, comme cela est très souvent le cas en modélisation du langage, nous travaillons avec des probabilités exprimées sous forme logarithmique.

Nous allons maintenant évaluer cette mesure simple sur deux corpus disjoints, dans le temps et donc dans leur contenu : *NewsFr* et *NewsFr2*. Le principe d'évaluation est très simple. On extrait de *NewsFr2* les messages ne contenant que le vocabulaire des groupes de l'arbre de thèmes construit sur *NewsFr*. En effet, comme *NewsFr2* est plus récent que *NewsFr*, il ne contient pas exactement la même arborescence de thèmes car de nouveaux thèmes sont apparus. De plus, certaines branches de l'arbre ont été redéfinies par une nouvelle structure. Cela a pour désagréable conséquence de faire disparaître certains termes de certains thèmes au profit d'autre thèmes. Alors, nous n'extrayons que le texte des groupes de *NewsFr2* modélisés par notre arbre.

Nous obtenons, par ce procédé, un ensemble de 1386 messages comprenant en moyenne une trentaine de mots. En appliquant la formule VII.5, nous obtenons un taux de bonne reconnaissance de thème résumé dans le tableau VII.1.

Thèmes	**Pseudos thèmes**	**Messages**	**Taux de reconnaissance thématique**
LB	oui	1386	69,83%

Tableau VII.1 : résultat de détection de thème

Ce résultat, inférieur à 70%, peut paraître faible. Certaines erreurs proviennent de thèmes qui sont proches même s'ils ne font pas partie de la même branche (comme 'science-fiction' et 'starwars'). Il est évident, dans ces cas là, que le choix d'un thème ou de l'autre n'est pas critique. Par contre, d'autres erreurs commises par le système sont vraiment importantes, comme la confusion entre un texte de 'fr.rec.cuisine' avec un thème très généraliste comme le pseudo thème 'fr.petites-annonces'. Cependant, nous verrons plus loin que cela n'implique pas forcement un problème lors de la phase de reconnaissance. Enfin, la majorité des erreurs ont été introduites sur des messages ne comprenant pas assez de mots pour que la détection de thème soit fiable. En conclusion, nous pensons que notre arbre de détection est tout à fait utilisable en reconnaissance de la parole pour tenter de limiter l'espace de recherche.

IV.2. Modification de l'algorithme *Tree-Forward*

Comme nous venons de le voir, notre détection de thème est plus efficace sur des messages longs. Or, notre but est l'intégration de ce détecteur au sein de la reconnaissance d'un seul énoncé. Pour pallier le manque de mots, nous avons eu l'idée d'intégrer notre détecteur de thème au sein de l'algorithme *Tree-Forward* de RAPHAEL. L'idée sous-jacente est d'employer, pour la détection, tous les mots contenus dans l'arbre de recherche de cet algorithme à chaque modification des hypothèses. Ainsi, comme plusieurs hypothèses contiennent des parties communes, en postulant que ces parties communes ne sont pas erronées, nous augmentons d'autant le poids thématique des mots présents plusieurs fois et ainsi, la chance de se trouver sur le bon thème.

Une nouvelle phase vient s'ajouter à l'algorithme *Tree-Forward*. À chaque modification des hypothèses, ce dernier calcule, avec tous les mots de toutes les hypothèses, le thème courant. Si le thème n'est pas le monde, à partir de ce point, toutes les probabilités du modèle générique de langage seront modifiées par la fonction de pondération que nous avons présentée plus tôt. Le thème courant évolue donc au fur et à mesure de l'apparition de nouvelles hypothèses.

Afin de profiter de la structure arborescente de notre arbre de thèmes et de son découpage logique en thèmes de plus en plus fins, nous avons mis au point plusieurs stratégies d'évaluation d'un thème. Le première, dite "tous les nœuds", est la plus coûteuse en temps de calcul. À chaque nouveau calcul de thème, on évalue tous les thèmes possibles de l'arbre. La seconde, "en avant", considère qu'il n'est possible, lorsque l'on est dans le thème *t*, que d'aller dans un sous-thème ou de rester dans le thème courant. La stratégie "avant arrière" autorise, en plus des règles de la précédente, à remonter au thème père, donc au nœud supérieur de l'arbre. La dernière stratégie n'évalue, à chaque tour, que les feuilles de l'arbre, n'intégrant ainsi aucun des pseudo-thèmes construits par fusion de sous thèmes.

IV.3. Évaluations

Nous allons maintenant évaluer notre algorithme modifié intégrant notre détection de thème. Dans un premier temps, nous évaluerons le potentiel de la méthode en fixant un thème connu à l'avance et en testant notre méthode de pondération. Par la suite, sur les données de l'évaluation *Aupelf*, nous donnerons les résultats obtenus avec nos différentes stratégies de navigation dans l'arbre de thèmes. Les résultats que nous présentons sont faits sur la base du corpus *WebFr*, car ils sont antérieurs à décembre 2000, date de collecte de *WebFr4*.

IV.3.a. Test de reconnaissance en fixant un thème

Ce test a pour but de valider notre méthode de repondération. Nous avons construit un modèle de langage incluant le vocabulaire de la tâche CSTAR (environ 2500 mots). Nous avons calculé le modèle de langage avec notre méthode des blocs minimaux sur la base d'un vocabulaire étendu à 30000 mots (59000 variantes phonétiques). Le modèle unigramme du thème CSTAR a été entraîné sur le corpus *réservations touristiques* qui, nous le rappelons, contient un ensemble de dialogues dans le domaine de la réservation touristique. Les résultats de reconnaissance sur le corpus sonore *CSTAR120*, avec et sans thème, sont présentés dans le tableau VII.2.

	Taux de reconnaissance
Modèle 30000 mots	56,30%
Modèle 30000 mots + thème CSTAR	61,20%

Tableau VII.2 : résultats de reconnaissance avec un seul thème fixé, le thème CSTAR

Le taux de reconnaissance avec ce modèle de langage non adapté à la tâche est de 56,30%, ce qui est nettement inférieur aux 78% obtenus avec le seul vocabulaire de la tâche CSTAR (cf. chapitre VI). Cependant, cette expérience montre qu'avec notre méthode de repondération par classe d'événements, nous pouvons avoir un gain significatif de presque 5%. De plus, parallèlement, nous avons réduit le temps de reconnaissance de 6% et cela en limitant l'espace de recherche de nos algorithmes par notre méthode de pondération. Nous avons donc atteint notre but, c'est-à-dire restreindre l'espace de recherche tout en augmentant la pertinence des résultats, sans pour autant réussir à combler totalement la perte en taux de reconnaissance induite par le passage d'un vocabulaire spécifique à un vocabulaire de 30000 mots. Cela démontre néanmoins la potentialité de la méthode.

IV.3.b. Test aveugle

Cette seconde phase de test à pour but d'évaluer, dans le cadre de la reconnaissance grand vocabulaire, l'intérêt de notre arbre de détection de thème, de son intégration dans l'algorithme de recherche de RAPHAEL et de notre technique de pondération thématique. Ce test est conduit sur les données de l'évaluation *Aupelf* avec un arbre de thèmes appris sur *NewsFr* qui compte 235 thèmes (179 feuilles plus 56 pseudo-thèmes). Le vocabulaire contenu dans les signaux de l'évaluation n'est pas complètement inclus dans notre arbre de thème. Tous les mots hors vocabulaire sont considérés comme ne faisant pas partie du thème courant. Le thème fixé au départ de chaque nouveau signal est le pseudo-thème 'fr', notre monde. Les résultats de cette évaluation sont donnés dans le tableau VII.3.

Stratégie de parcours de l'arbre	**Taux de reconnaissance**
sans thème	51,00%
tous les nœuds	43.50%
en avant	46.10%
avant arrière	46.10%
feuilles seulement	54.10%

Tableau VII.3 : résultats de l'évaluation aveugle sur les données Aupelf

Nous tenons à rappeler au lecteur que 51% ne représente pas notre meilleur score sur la tâche *Aupelf*, mais celui que nous obtenons en utilisant *WebFr* pour l'apprentissage de notre modèle de langage. La première remarque est que la méthode la plus coûteuse, l'évaluation de tous les nœuds, est celle qui le moins bon résultat en taux de reconnaissance.

Pourtant, le gain en temps de reconnaissance en utilisant cette méthode est de l'ordre de 9%. Malgré son coût, cette méthode permet de limiter l'espace de recherche des algorithmes de reconnaissance. Les deux stratégies "en avant" et "avant arrière" fournissent strictement les mêmes résultats. En fait, dans notre évaluation, il n'y a jamais eu de retour au thème supérieur, donc la méthode "avant arrière" est identique à la méthode "en avant". Ces deux méthodes donnent un résultat significativement supérieur à celui de la stratégie exhaustive. De plus, le gain en temps de reconnaissance est de 11%, ce qui prouve bien la limitation de

l'espace de recherche. La meilleure stratégie est celle consistant à évaluer, à chaque fois que cela est possible, toutes les feuilles de l'arbre de thèmes. Il y a cette fois, gain en taux de reconnaissance, de 51 à 54,10%, et 13% de temps de calcul en moins.

Il est difficile de juger, avec cette seule expérience, notre méthode et ses résultats. Les problèmes peuvent être dus à notre fonction de pondération. Nous pouvons aussi penser que cela provient du fait que les feuilles correspondent à des données réelles alors que les nœuds sont, dans leur très grande majorité, construit par fusion de thèmes. Nous pourrions alors critiquer notre méthode de construction de pseudo-thèmes. Toutefois, cette expérimentation prouve qu'il est possible de déterminer automatiquement le thème d'un seul énoncé avec des thèmes construits à partir des *newsgroups*. Nous obtenons non seulement un gain en taux de reconnaissance, mais aussi une limitation de l'espace de recherche résultant en un gain de temps de reconnaissance.

Conclusion

Notre méthode, au vu des résultats d'expérimentations que nous obtenons, n'est pas dénuée de sens. Nous avons montré que notre fonction de pondération peut s'avérer utile si l'on connaît le thème de la discussion (expérience en fixant le thème CSTAR). De plus, nous avons aussi montré que l'utilisation du vocabulaire contenu dans toutes les hypothèses, lors de la phase de reconnaissance, peut conduire à une réduction de l'espace de recherche et à un gain de reconnaissance (expérience en aveugle). Nous travaillons actuellement à l'amélioration de ces résultats, en particulier en réalisant des prétraitements sur les thèmes présents dans l'arbre pour regrouper certaines branches.

Pour les perspectives à plus long terme des ces travaux, nous nous intéressons particulièrement à des techniques comme l'analyse à sémantique latente [Bellagarda 01b] (*Latent Semantic Analysis*), elle aussi tirée des travaux en Recherche d'Informations, pour l'adapter dans le contexte de notre étude. En deux mots, cette technique associe les probabilités d'un modèle de langage statistique avec une information sur les cooccurrences de mots dans un domaine particulier. Par rapport à notre méthode, qui ne gère que des classes d'événements, celle-ci est beaucoup plus proche des méthodes *trigger*. Nous étudions actuellement sa mise en pratique au sein de notre détecteur de thème.

Chapitre VIII : Adaptation de notre méthode pour la définition de corpus sonore

Chapitre VIII : Adaptation de notre méthode pour la définition de corpus sonore

Présentation du chapitre

RAPHAEL, notre système de reconnaissance, fournit de bons résultats comme nous l'avons vu dans les chapitres précédents. Cependant, son module acoustique n'est entraîné que sur une dizaine d'heures de parole ce qui est insuffisant d'après la littérature. Nous avons donc décidé d'enregistrer notre propre corpus sonore pour avoir une plus grande base d'entraînement et ainsi augmenter encore ses performances. La difficulté d'une telle tâche est la définition d'un ensemble de phrases à faire prononcer par les locuteurs pour l'enregistrement. Comme notre filtre de blocs minimaux est capable de générer, à partir de pages Web, des corpus ne contenant que des phrases complètes, nous avons décider d'utiliser ces données pour nous aider dans notre travail.

Nous présentons dans ce chapitre, notre adaptation de la méthode des blocs minimaux dans le cadre de la définition d'un ensemble d'énoncés pour l'enregistrement d'un corpus sonore. Nous verrons comment nous pouvons limiter le travail de relecture à faire, en utilisant nos outils de modélisation du langage. Enfin, nous présenterons le corpus BRAF-100 (Base pour la Reconnaissance Automatique du Français - 100 locuteurs), que nous avons construit avec cette technique puis enregistré au sein de notre laboratoire.

I. Introduction

Nous avons présenté, dans la seconde partie de notre manuscrit, nos travaux sur la modélisation du langage. Cependant, comme nous l'avons vu dans le second chapitre, le premier module d'un système de reconnaissance de la parole, le module acoustique, est aussi très important. Pour son entraînement, il est nécessaire de posséder un corpus sonore de taille suffisante pour obtenir de bons modèles HMMs. À l'heure actuelle, nous n'avons que des modèles appris sur BREF-80 [Lamel et al. 01] qui ne compte qu'une dizaine d'heures de parole. Nous travaillons aussi sur l'utilisation de BREF-TOTAL qui est lui beaucoup plus gros. Cependant, le problème de ces deux corpus est la lenteur relative des locuteurs. En effet, si l'on écoute ces signaux, qui correspondent à des extraits lus du journal « Le Monde », on se rend vite compte que le débit de parole est inférieur à celui d'une personne qui dialogue avec une autre. Or, nous travaillons à la reconnaissance de la parole spontanée. Même si les modèles HMMs dits de Bakis permettent une tolérance à de la parole rapide (cf. Chapitre II, section III.2), si la différence entre le débit de parole de l'utilisateur et les données d'apprentissage est trop importante, cela ne sera pas suffisant pour que le système de reconnaissance fonctionne correctement.

Nous avons alors lancé le projet d'enregistrer un corpus proposant un débit de parole supérieur aux corpus disponibles. Cependant, il fallait pour cela définir un ensemble d'énoncés à faire prononcer par les locuteurs. L'idée d'utiliser nos corpus de documents extraits de la Toile pour obtenir cet ensemble est alors apparue comme une évidence. Le postulat de ce travail était qu'il était possible d'utiliser les outils que nous avions mis au point pour le calcul de modèle de langage dans ce but [Vaufreydaz et al. 00].

II. Génération d'énoncé pour l'enregistrement de corpus sonore

II.1. Méthode d'obtention

Le schéma suivant présente la méthode de filtrage utilisée pour obtenir notre ensemble d'énoncés depuis le corpus *WebFr*.

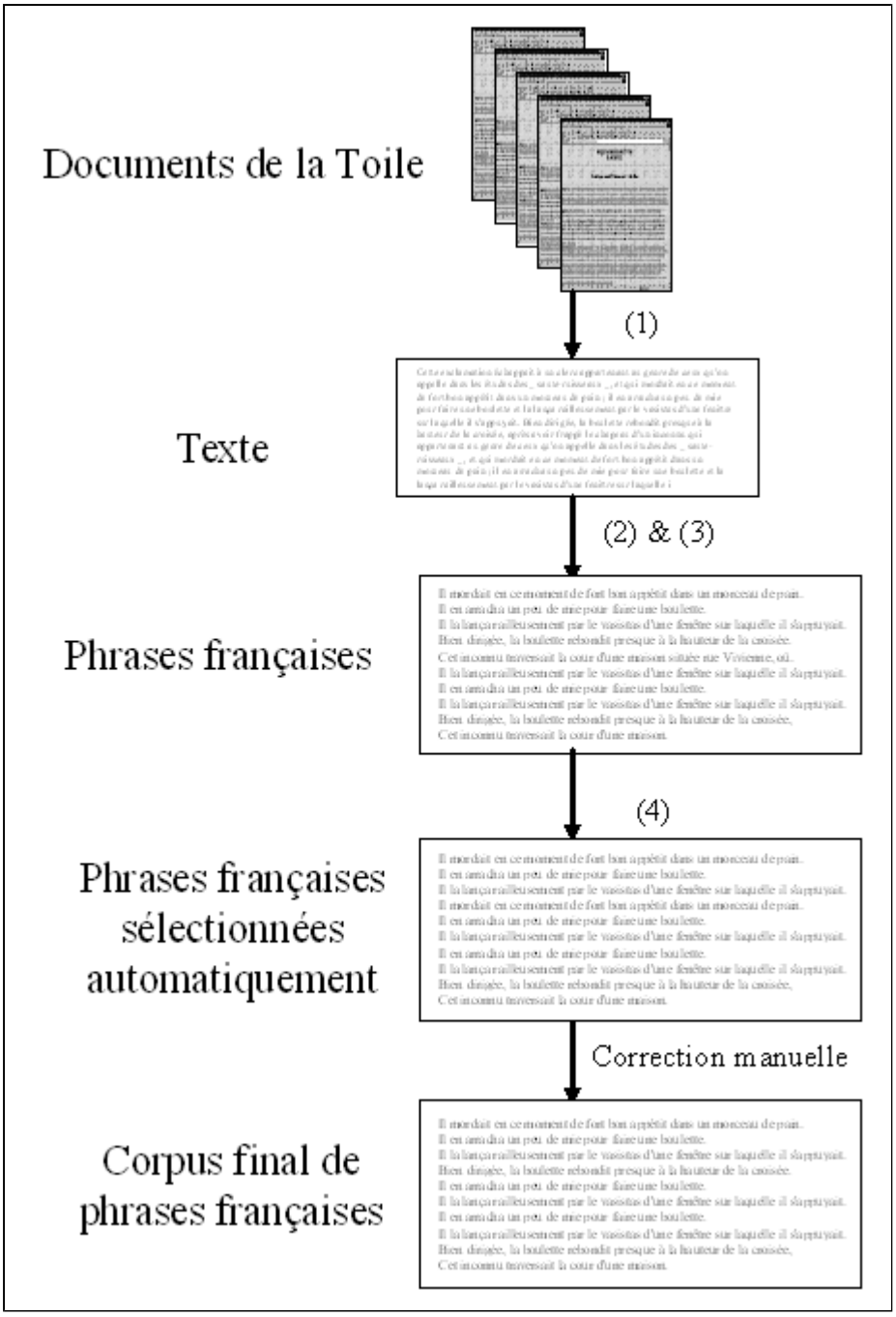


Figure VIII.1 : synoptique de la méthode d'extraction d'énoncés

Le principe d'extraction des énoncés à partir des documents de la Toile est, au début, le même que pour les modèles de langage. À partir du corpus *WebFr* (nous ne possédions pas les autres au moment où nous avons réalisé ce travail), nous obtenons un texte impur. Nous filtrons ce texte, avec un vocabulaire maximal de 400000 formes fléchies du français, obtenu par union des lexiques d'ABU [Web 08] et de BDLEX [Perennou et al. 87], par la méthode des blocs minimaux (1). Cette extraction est accompagnée de la transcription des nombres en contexte pour les dates, les heures, etc. Cela élimine la variété des prononciations possibles d'une même représentation numérique par différents locuteurs, ce qui était le cas dans BREF-80 où ce procédé n'a pas été employé. Ainsi, pour la prononciation de "20,20 francs" par

exemple, certains locuteurs ont dit "vingt francs vingt centimes" d'autres "vingt virgule vingt francs", ce qui pose des problèmes lors de l'alignement de phonèmes sur les signaux de parole.

Le second filtre (2) est notre filtre de blocs minimaux. Il utilise notre lexique de 400000 formes fléchies et ne génère que des phrases complètes. Ce choix de paramètres est motivé par le fait qu'il est difficile pour un locuteur de prononcer des phrases qui sont des non-sens ou qui comportent des erreurs. Effectivement, cela introduit une surcharge cognitive de nature à perturber la production de parole, ce qui peut se traduire par exemple par une variation de la prosodie [Joux 98]. Or, un bloc minimal ne représente pas une entité complète et peut, par là-même, entraîner ce genre de problème. Notre corpus d'énoncés comporte après ce filtre 70759 phrases.

La troisième étape (3) utilise un ensemble de règles simples pour réaliser un filtrage des phrases obtenues jusqu'à cette étape et supprimer certains types de phrases :

- tout d'abord, on enlève les phrases en double car nous souhaitons connaître à l'avance les énoncés communs à différents locuteurs. Nous avons donc choisi de fixer l'élément commun et de ne pas garder de doublons par ailleurs. Cet élément commun est un texte de 5 phrases et 99 mots extrait de « La science et l'hypothèse » d'Henri Poincaré.
- les phrases comportant plus d'un point, c'est-à-dire contenant des acronymes ou des expressions comme "etc.", sont supprimées car il peut aussi y avoir différentes manières de les prononcer.
- les phrases contenant des épellations sont enlevées.
- les doublons de mots, qui sont souvent des signes d'erreur, hormis dans des cas comme "nous nous".
- nous n'avons enfin gardé que les phrases de plus de 15 mots afin de limiter le nombre d'énoncés à faire prononcer par les locuteurs, 15 étant un seuil fixé empiriquement.

Après ce troisième filtre, nous obtenons un ensemble de 47482 énoncés.

Nous allons maintenant mettre en place un filtrage plus avancé (4). Dans un premier temps, nous avons utilisé un lemmatiseur [Schmid 94] sur le texte d'un corpus propre, le corpus *Grace* [Web 06] qui contient des extraits du journal « Le Monde ». Nous avons calculé un modèle n-classe n'incluant que les probabilités de cooccurrence de classes. Nous avons alors lemmatisé nos énoncés. Le rejet de phrases est alors fait selon deux critères. Si une phrase contient un enchaînement de 3 classes que nous n'avons pas dans notre modèle n-classe, elle est supprimée. Ce critère est donc simplement une analyse statistique de la structure syntaxique de nos phrases. Notre ensemble d'énoncés contient maintenant 43279 éléments.

Nous visions un corpus d'une vingtaine d'heures. Sachant que nos phrases comportent au moins 15 mots et que nous estimions empiriquement qu'il faut 6 secondes pour énoncer chacune d'elle, le nombre de phrases dont nous avons besoin était d'environ 12000. Nous avons donc, pour chacun de nos énoncés, calculé la perplexité de notre modèle n-classe grâce aux outils de [Rosenfeld 95]. Dans ce cas, la perplexité est utile car elle correspond bien au pouvoir de prédiction du modèle pour chacune de nos phrases sans notion d'acoustique,

contrairement à la reconnaissance de la parole. Inversement, on peut aussi voir la perplexité d'un modèle de langage sur une phrase comme la justesse de la phrase par rapport au modèle. Ainsi, plus la perplexité est faible, plus la phrase correspond au modèle. En fixant manuellement le seuil maximum de perplexité à 12, nous obtenons un corpus de 12239 phrases.

Arrivé à ce stade, il ne reste plus qu'une seule solution pour finaliser le travail automatique : une relecture manuelle des énoncés. Cela a été fait par plusieurs chercheurs du laboratoire en utilisant une interface de travail collaboratif via un navigateur Web. À cause de problèmes techniques, notre moteur de base de données ne gérant pas les champs textuels trop longs, nous n'avons pu proposer à la correction que 11262 énoncés. Parmi ceux-ci 792 furent jugés comme incorrigibles par les relecteurs. Ils étaient majoritairement constitués de suite de mots sans aucun sens. Encore une fois, cela illustre bien qu'un modèle statistique de type n-classe, mais cela est généralement vrai pour tous les modèles statistiques, est capable de donner un bon score à une séquence totalement incorrecte. Au total, sur les 10470 énoncés restants, 74% n'ont pas été corrigés du tout. Si l'on considère que le changement de ponctuation ne constitue pas une correction majeure, ce taux atteint 77%. Les autres 23% ont été corrigés pour obtenir des phrases correctes à faire prononcer par les locuteurs.

II.2. Caractéristiques phonétiques des énoncés obtenus

À ce point, nous possédons un corpus de phrases prêtes à être prononcées par des locuteurs. Cependant, le but premier de ce corpus est l'entraînement des modules acoustiques de reconnaissance de la parole. Nous devons donc nous assurer que celui-ci est représentatif du français, d'un point de vue acoustique, pour qu'il soit considéré de bonne qualité. Nous avons alors phonétisé l'ensemble de nos phrases à l'aide du programme *text2phon* du système de synthèse MBROLA [Dutoit et al. 96]. Nous comparons, sur le graphique VIII.2, nos résultats avec ceux présentés par [Combescure 81].

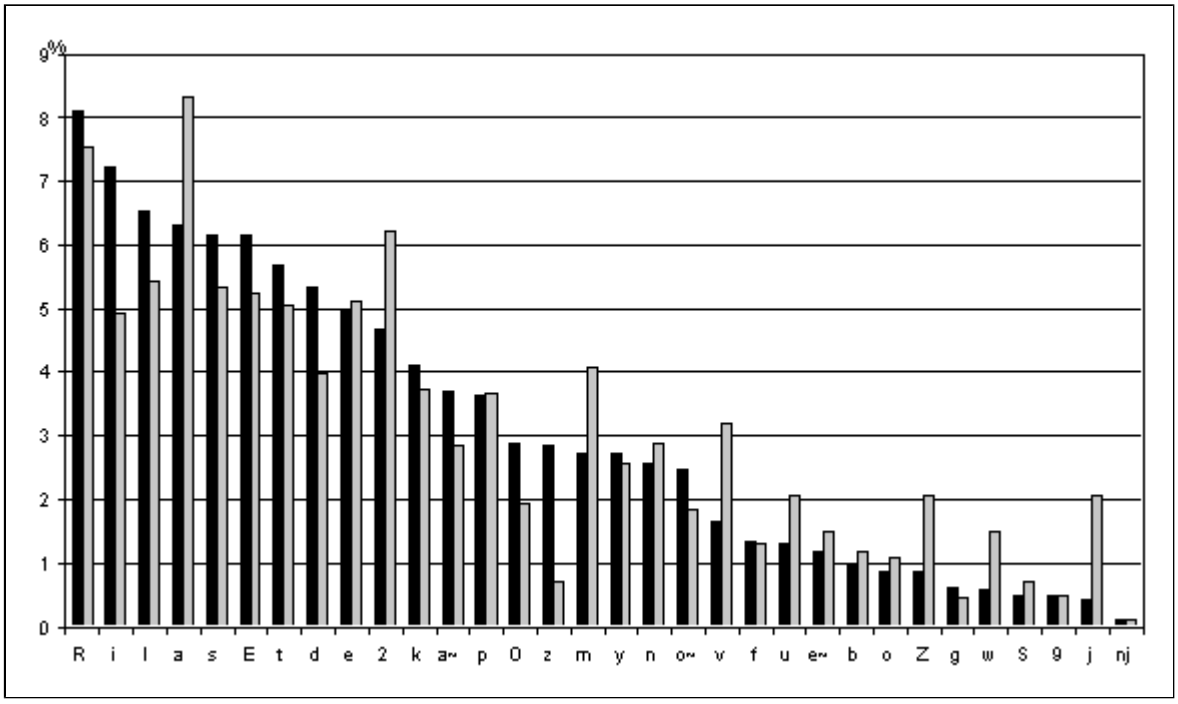


Figure VIII.2 : comparaison entre la représentation phonétique dans nos énoncés et les valeurs de référence de Combescure

Le graphique précédent présente la répartition en pourcentage de chacun des phonèmes de notre corpus (en noir) comparée à celle fournie par Combescure (en gris). Nous voyons que les deux sont très proches. Le coefficient de corrélation entre ces deux séries est de 0,89. Notre corpus est donc équilibré phonétiquement par rapport au français tel qu'il avait été étudié par Combescure.

III. BRAF-100, le corpus final

III.1. Phase d'enregistrement

L'enregistrement de BRAF-100 (Base pour la Reconnaissance Automatique du Français - 100 locuteurs) a été effectué au cours de l'année 2000. Le système d'acquisition utilisé est EMACOP [Vaufreydaz et al. 98a]. Cet ensemble de logiciels permet l'acquisition, en mode client/serveur, de plusieurs locuteurs en même temps. De plus, il ne nécessite pas d'appuyer sur un bouton pour enregistrer, mais détecte, grâce à une variante du calcul de l'énergie du signal, lorsque le locuteur parle. Enfin, le système exporte le corpus final en respectant le standard SAM [Tomlison 91], ce qui permet un échange aisé de données entre les laboratoires ainsi que leur portabilité, sur plusieurs systèmes d'exploitation.

L'enregistrement des locuteurs a été fait dans des conditions environnementales de type bureau, et non dans une chambre sourde comme c'est le cas pour certains autres corpus. La chaîne d'acquisition était constituée d'un microphone Sennheiser HMD 410-6 avec un pré-amplificateur PREFER MB-7 et d'un filtre qui éliminait les fréquences inférieures à 60 Hz. La fréquence d'échantillonnage des signaux obtenus est de 16000 Hz avec une quantification linéaire sur 16 bits.

III.2. Organisation du corpus

Le corpus est divisé en deux parties distinctes : la partie test et la partie apprentissage. Sur les 100 locuteurs, la partie apprentissage comprend 90 locuteurs, la partie test les 10 autres. La découpe en deux de notre corpus a été faite de manière à conserver la représentation phonétique de chacune des sous-parties en choisissant, de manière automatique avec un programme que nous avons écrit, les énoncés que chacun des locuteurs devraient prononcer. Chaque locuteur a prononcé de 102 à 105 phrases, comportant en moyenne 26 à 28 mots, plus le texte extrait de « La science et l'hypothèse » d'Henri Poincaré. Ce passage commun peut être utile dans le cadre de recherches sur l'identification du locuteur dépendante du texte par exemple.

Au final, BRAF-100 est un corpus sonore comprenant une partie test de 2h40 et une partie apprentissage de 25h17. Toutes les transcriptions textuelles associées aux signaux ont été vérifiées manuellement, ce qui permet de faciliter le travail d'alignement nécessaire pour tout travail sur des données sonores dans le domaine de la reconnaissance de la parole.

III.3. Statistiques

III.3.a. Sur les locuteurs

Lors de la phase d'enregistrement, nous avons cherché à avoir une répartition homme/femme équilibrée, ce qui permet d'obtenir des modèles acoustiques capables de reconnaître aussi bien des voix de femmes que des voix d'hommes. Chaque partie du corpus, et donc le corpus total, comprend 50% de femmes et 50% d'hommes. Nous avons aussi tenté d'avoir une répartition des locuteurs en fonction de leur âge la plus complète possible. Le plus jeune locuteur a 15 ans et le plus âgé 63. La répartition par âge et par sexe des locuteurs dans chacune des parties est donnée dans le tableau VIII.1.

Âge	**Partie test**		**Partie apprentissage**	
	femme	**homme**	**femme**	**homme**
≤ 20	0	0	2	1
21 - 30	3	3	26	30
31 - 40	1	1	8	8
41 - 50	1	0	6	5
51 - 60	0	1	6	5
> 60	0	0	2	1

Tableau VIII.1 : répartition des locuteurs par âge et par sexe

Comme le montre bien le tableau précédent, il a été beaucoup plus difficile de répartir les locuteurs, selon les âges et le sexe, au sein du corpus. Cela provient du fait que les personnes s'inscrivant pour s'enregistrer le faisaient au fur et à mesure et qu'il était impossible de prévoir comment cela allait évoluer.

III.3.b. Sur le contenu des signaux

Hormis les informations sur les locuteurs, il est important d'avoir une connaissance sur le contenu des signaux. Le tableau ci-après donne des informations sur le contenu textuel des signaux.

	Nombre de phrases	**Nombre de mots**	**Taille du vocabulaire**
test	1026	27926	6035
apprentissage	9438	257066	19796
Total	10464	284992	20732

Tableau VIII.2 : information textuelle contenue dans les signaux

Le lecteur notera que le nombre total de phrases est inférieur au nombre d'énoncés que nous avions produits. En effet, 6 signaux ont été supprimés lors de la phase d'écoute et de correction des transcriptions textuelles, car ils ont été jugés trop mal prononcés.

Les dernières informations que nous donnerons ici sur BRAF-100 concernent la vitesse d'élocution des locuteurs. Le tableau VIII.2 présente la durée totale d'une session, la durée moyenne de chacune des phrases, du passage commun, et le débit en nombre de mots par seconde. Ces informations sont données, pour chaque partie, pour le locuteur le plus lent, le plus rapide et la moyenne sur tous les locuteurs. Il n'existe aucun lien vertical entre les informations présentées. Ainsi, la session la plus courte n'est pas celle qui possède le débit en mots/seconde le plus rapide. Cela dépend du nombre de mots dans la session.

	test			apprentissage		
	min.	**max.**	**moy.**	**min.**	**max.**	**moy**
Durée totale d'une session (en s)	869	1081	947	818	1438	1011
Durée moyenne d'une phrase par locuteur (en s)	7,77	10,23	8,93	7,44	13,06	9,21
Durée de la partie commune (en s)	30,21	40,60	35,54	28,67	54,27	35,45
Nombres de mots/secondes	2,70	3,41	3,05	1,99	3,67	2,95

Tableau VIII.3 : information de durée et de débit en mots/seconde sur les signaux

Nous pouvons voir, dans le tableau précédent, que les informations sont très similaires pour les parties test et apprentissage. Il n'est donc pas aberrant d'utiliser la partie test pour juger d'un apprentissage acoustique mené sur la partie apprentissage.

Conclusion

Notre méthode d'extraction et de filtrage de texte, basée sur des blocs minimaux, a pu être adaptée avec succès à la définition d'un ensemble de phrases pour l'enregistrement de corpus sonores. L'avantage de cette méthode est le gain de temps par rapport à une définition manuelle d'énoncés. En effet, 75% des énoncés générés n'ont pas à être modifiés. De par la richesse et la variété des textes que l'on peut trouver sur la Toile, il semble que notre démarche aboutisse, de plus, à un ensemble de phrases ayant de bonnes propriétés phonétiques. Le corpus BRAF-100 sera bientôt disponible pour les autres équipes de recherche via l'association ELRA (*European Language Resources Association*) [Web 14].

Conclusions et perspectives

Conclusions et perspectives

La reconnaissance de la parole, malgré des performances allant croissant, n'est pas un problème résolu, en particulier vis à vis de la modélisation du langage. Notre thèse contribue à l'amélioration des techniques de modélisation statistique du langage par l'apport d'une technique de filtrage des documents en provenance d'Internet, dite par bloc minimaux, et par l'adaptation des outils standard de calcul de modèles de langage pour l'apprentissage sur de telles données.

Les avantages de l'utilisation d'Internet pour constituer des corpus d'apprentissage sont multiples. Le premier concerne évidemment la taille des corpus qu'il est possible d'en extraire. Nous avons vu qu'avec *WebFr4*, notre dernier corpus, et pour un vocabulaire d'environ 20000 mots, nous obtenons un corpus d'apprentissage pour nos modèles statistiques d'un milliard et demi de mots. Cela est très largement supérieur à la taille des corpus usuels, extraits de journaux par exemple. Le second avantage des documents en provenance d'Internet est leur capacité à fournir des formes de type dialogique difficile à trouver dans les corpus habituellement employés pour calculer des modèles de langage. Fernando Pereira le soulignait d'ailleurs à la conférence *Human Language Technology* en 2001 dans son allocution intitulée « Web Language is natural language » : « Structured documents have not only many of the same kinds of ambiguity that we know from natural languages, but also create new ones that will keep computational linguists busy for the foreseeable future. ». Le dernier avantage d'Internet concerne la couverture linguistique de ces corpus. Celle-ci est très large et va croissant, grâce à la diversité des personnes ayant accès au réseau, comme nous l'avons montré dans le chapitre IV. De plus, nous pensons que sur Internet, il se produit le même phénomène que celui décrit par Walter dans [Walter 97] et dans [Walter 98], c'est-à-dire l'intégration de mots nouveaux ou venant d'autres langues, mais à un rythme beaucoup plus rapide du fait de la vitesse de diffusion de l'information sur ce média. Ce fut récemment le cas avec le mot *startup* et son équivalent français *jeune pousse* désignant les sociétés naissantes des nouvelles technologies. Cette évolution rapide permet de prendre en compte tous ces mots dans un système de reconnaissance. Il est alors suffisant de collecter un nouveau corpus, d'ajouter les nouveaux mots au vocabulaire et de recalculer, grâce à nos outils, un nouveau modèle de langage. Il est même tout à fait envisageable de

suivre automatiquement l'évolution du vocabulaire présent sur Internet, pour construire, chaque semaine ou chaque mois par exemple, un nouveau modèle de langage reflétant au mieux le vocabulaire le plus utilisé du moment.

L'intérêt d'Internet pour la modélisation du langage commence à se répandre dans la communauté scientifique. Nous pouvons citer la méthode décrite dans [Zhu et al. 01]. Celle-ci consiste non pas à collecter des documents, comme nous le faisons, mais à interroger des moteurs de recherche sur la Toile avec des trigrammes. Ensuite, le nombre de pages contenant ces trigrammes, renvoyé par le moteur, est utilisé dans le calcul des probabilités associées à ces trigrammes.

Si l'on s'intéresse maintenant aux résultats de reconnaissance que nous obtenons grâce à nos outils, nous observons que ceux-ci sont intéressants sur plusieurs points. Premièrement sur une tâche limitée comme la tâche type *CSTAR* (2500 mots), il est possible d'atteindre un taux de reconnaissance de 88% sur des énoncés en langage naturel. Sur une tâche similaire, mais en générant un modèle de langage de 20000 mots (tâche *Nespole!*), les résultats sont de 80% avec une couverture linguistique très large. Conformément à nos attentes, la taille des corpus extraits d'Internet et leur variété nous ont permis de prendre en compte des phénomènes propres aux énoncés en langage naturel comme les répétitions, les hésitations, etc. Si l'on considère maintenant le travail sur les données d'évaluation de l'AUPELF, notre taux de reconnaissance est d'environ 62%. Ce résultat reste dans le même ordre de grandeur que ceux des systèmes de reconnaissance ayant participé à l'évaluation. Bien que ce résultat soit améliorable (nous n'avons réalisé aucune adaptation ni acoustique ni sur le modèle de langage), nous pouvons affirmer que les corpus extraits entièrement automatiquement d'Internet sont presque aussi bons que les textes jumeaux du journal « Le Monde » fournis aux participants de l'évaluation. Nous rappelons aussi que notre module acoustique n'est entraîné que sur 10 heures de parole.

En ce qui concerne l'intégration d'un module de détection de thème basé sur une arborescence de *newsgroups*, nous ne sommes actuellement qu'au début de nos recherches. Cependant, nous avons pu montrer l'intérêt de notre approche dans deux tests différents. Le premier, en fixant le thème pour valider notre méthode de pondération thématique, nous a permis d'obtenir un gain de 5% en taux de reconnaissance avec, parallèlement une réduction de 6% du temps de reconnaissance. Notre approche permet donc de limiter l'espace de recherche des algorithmes de reconnaissance tout en augmentant leur performance en terme de taux de reconnaissance. Le second test propose un mode aveugle, c'est-à-dire en ne connaissant pas à l'avance le thème, avec un parcours non supervisé d'un arbre contenant une hiérarchie de thèmes. Nous avons mesuré de taux d'identification de thèmes égal à 69% en utilisant un arbre thématique construit à partir des *newsgroups* français. Nous avons modifié l'algorithme de reconnaissance de notre système pour qu'il prenne en compte cet arbre de thèmes. En testant différentes stratégies de recherche d'un thème au sein de l'arbre, nous avons mesuré, pour la meilleure d'entre elles, un gain significatif en taux de reconnaissance de 3% s'accompagnant d'un gain en temps de reconnaissance de 13%.

Nos méthodes et outils ont aussi montré leur adéquation pour la définition d'un ensemble de phrases à recueillir pour l'enregistrement de corpus sonore. En ne modifiant que

très légèrement nos algorithmes, nous avons pu définir, avec un minimum d'effort, un ensemble de plus de 10000 phrases phonétiquement équilibrées, en n'en corrigeant manuellement que 25%. Cela a très nettement réduit le travail nécessaire. Le corpus résultant de ces travaux, BRAF-100, devrait nous permettre d'accroître la robustesse de notre module acoustique.

Nous pouvons affirmer qu'Internet est bien une source possible de données pour la modélisation du langage. Nous avons validé nos hypothèses concernant la taille et la qualité de ces données. Nos outils de collecte et de filtrage permettent d'obtenir aisément et très rapidement des modèles de langage de bonne qualité en réduisant l'effort à fournir au strict minimum, c'est-à-dire la définition du vocabulaire de base de l'application. Comme nous l'avons vu, il est aussi possible de l'utiliser pour d'autres applications, par exemple comme la détection de thème.

Les perspectives à court terme de nos travaux concernent l'amélioration de nos techniques de détection de thème et leur intégration au sein des algorithmes de reconnaissance. Les résultats que nous obtenons valident notre méthode de pondération, mais nous continuons nos recherches dans le but d'améliorer notre système. Dans le même temps, nous allons réaliser le portage et l'évaluation de nos techniques de modélisation du langage dans d'autres langues. Nous avons déjà commencé à travailler sur l'allemand en collaboration avec l'université de Karlsruhe, ce qui nous permettra de réaliser des évaluations sans avoir à construire nos propres modèles acoustiques. L'avantage de cette langue est aussi de fournir des difficultés supplémentaires par rapport au français. En effet, les travaux présentés dans ce manuscrit utilisent une liste maximale de vocables français pour le filtrage des textes. En allemand, il existe des règles de compositions de mots, par exemple pour les nombres, qui nécessitent une adaptation de nos outils de filtrage. Nous allons aussi porter nos outils en espagnol mexicain en collaboration avec l'INAOE (*Instituto Nacional de Astrofisica Óptica y Electronica*), un laboratoire de Puebla au Mexique, dès le début de l'année 2002. Enfin, nous pensons aussi expérimenter nos méthodes en langue anglaise avec la collaboration de Carnegie Mellon University à Pittsburgh aux USA. Ces recherches devraient nous permettrent d'acquérir une connaissance multilingue de la modélisation du langage basée sur Internet.

En ce qui concerne les perspectives à plus long terme, divers axes de recherche visent à la définition d'une modélisation acoustique multilingue (comme au sein du projet GlobalPhone de CMU/Karlsruhe) mais peu considèrent le multilinguisme comme un enjeu de la modélisation du langage. Pourtant, nous pensons que l'avenir passe probablement par cette étape. Sur Internet, il est possible de trouver de très nombreux textes présents en plusieurs langues. Notre but serait alors d'arriver à mettre en correspondance ces textes pour pouvoir obtenir un modèle multilingue de langage. Cependant, cela implique de définir un nouveau formalisme de représentation de nos modèles. Effectivement, la combinatoire d'un modèle trigramme français est déjà très élevée. Si l'on intègre plusieurs langues dans le même modèle, il est évident que la taille de celui-ci va croître de manière exponentielle. La forme de ce formalisme reste à déterminer et dépendra fortement des connaissances de plus haut niveau sur lesquelles il sera fondé. Cependant, nous pouvons imaginer qu'il pourra être construit autour d'un langage pivot sémantique autorisant, pour chaque sens et dans plusieurs langues,

diverses constructions grâce à un sous modèle de langage statistique. Deux hypothèses concernant ce pivot sont envisageables à l'heure actuelle : l'IF [Levin et al. 98], le langage pivot utilisé pour CSTAR et *Nespole!* ou UNL [Sérasset et al 99] (*Universal Networking Language*). L'adéquation de chacun d'eux aux connaissances linguistiques extraites de l'Internet permettra de choisir le plus approprié.

À ce stade, nous pourrons alors envisager l'utilisation de nos modèles de langage multilingues dans d'autres applications que la reconnaissance de la parole. Ceux-ci pourraient par exemple être utiles pour construire des systèmes de compréhension ou de traduction automatique de la parole.

Références bibliographiques

Publications scientifiques

[Akbar et al. 98] Akbar M., Caelen J., *Parole et traduction automatique : le module de reconnaissance automatique RAPHAEL*, COLING-ACL'98, Montréal (Quebec, Canada), volume I, pp. 36-40, août 1998.

[Antoine et al. 01] Antoine J.Y., Goulian J., *Word order variations and spoken man-machine dialogue in French : a corpus analysis on the ATIS domain*, Corpus Linguistics'2001, Lancaster (Angleterre), 2001.

[Aubry et al. 00] Aubry M., Cellier F., *Reconnaissance vocale sur plate-forme télécom Hewlett Packard*, rapport de stage de 3ème année de l'ENS d'Électronique et de Radioélectricité de Grenoble, juin 2000.

[Bacchiani et al. 01] Bacchiani M., Hirschberg J., Rosenberg A., Whittaker S., Hindle D., Isenhour P., Jones M., Stark L., Zamchick G., *SCANMail: Audio Navigation in the Voicemail Domain*, Human Language Technology conference, San Diego California (USA), mars 2001.

[Bahl et al. 83] Balh L.R., Jelinek F., Mercer L., *A Maximum likelihood approach to continuous speech recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume PAMI-5, pp. 179-190, mars 1983.

[Baum et al. 66] Baum L.E, Petrie T., *Statistical inference for probabilistic functions of finite state Markov chains*, Annals of Mathematical Statistics (37), pp. 1554-1563, 1966.

[Baum 72] Baum L.E, *An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes*, Inequalities, numéro 3, pp. 1-8, 1972.

[Beaujard et al. 99] Beaujard C., Jardino M., *Language Modelling Based on Automatic Word Concatenations*, Eurospeech'99, pp. 1563-1566, Volume 4, Budapest (Hongrie), septembre 1999.

[Bellagarda 01a] Bellagarda J., *An Overview of Statistical Language Adaptation*, ITR-Workshop on Adaptation Methods For Speech Recognition, pp. 165-174, Nice Sophia-Antipolis (France), août 2001.

[Bellagarda 01b] Bellagarda J., *A New Approach to the Adaptation of Latent Semantic Information*, ITR-Workshop on Adaptation Methods For Speech Recognition, pp. 191-194, Nice Sophia-Antipolis (France), août 2001.

[Bergamini 00] Bergamini C., *Modèles acoustiques dépendants du contexte pour la reconnaissance de la parole*, DEA Informatique Système et Communication de Université Joseph Fourier de Grenoble, 2000.

[Besacier et al.] Besacier L., Vaufreydaz D., Grassi S., *Remote Recognition: the speech coding problem*, soumis à revue pour IEEE Transaction on Speech and Audio Processing.

[Besacier et al. 01a] Besacier L., Blanchon H., Fouquet Y., Guilbaud J.P., Helme S., Mazenot S., Moraru D., Vaufreydaz D., *Speech Translation for French in the NESPOLE! European Project*, Eurospeech'01, pp. 1291-1294, AAlborg (Danemark), septembre 2001.

[Besacier et al. 01b] Besacier L., Bergamini C., Vaufreydaz D., Castelli E., *The Effect of Speech and Audio Compression on Speech Recognition Performance*, pp. 301-306, Cannes (France), octobre 2001.

[Bigi 00] Bigi B., *Contribution à la modélisation du langage pour des applications de recherche documentaire et de traitement de la parole*, thèse de l'université d'Avignon et des pays de Vaucluse, septembre 2000.

[Bimbot et al.] Bimbot F., El-Bèze M., Igounet S., Jardino M., Smaïli K., Zitouni I., *An Alternative Scheme for Perplexity Estimation and its Assessment for the Evaluation of Language Models*, Journal of Computer Speech and Language, Academic Press, à paraître.

[Bimbot et al. 97] Bimbot F., El-Bèze M., Jardino M., *An alternative scheme for perplexity estimation*, ICASSP'97, Munich (Allemagne), 1997.

[Bonafonte et al. 96] Bonafonte A., Marino J.B., *Language modeling using x-grams*, ICASSP'96, pp. 394-397, 1996.

[Brown et al. 90] Brown P., Cocke J., Della Pietra S., Della Pietra V., Jelinek F, Lafferty J., Mercer R., Roossin P., *A statistical approach to machine translation*, Computational Linguistics, 16(2):79-85, juin 1990.

[Burger et al. 01] Burger S., Besacier L., Coletti P., Metze F., Morel C., *The Nespole! VoIP Dialogue Database*, Eurospeech'01, pp. 2043-2046, AAlborg (Danemark), septembre 2001.

[Caelen 85] Caelen J., *Space/Time Data-Information in the A.R.I.A.L. Project Ear Model*, Speech Communication 4, Elsevier Science Publishers B.V., pp. 163-179, North-Holland, 1985.

[Cattoni et al. 01] Cattoni R., Federico M., Lavie A., *Robust Analysis of Spoken Input Combining Statistical and Knowledge-Based Information Sources*, ASRU'01, à paraître, Trento (Italie), décembre 2001.

[Cerf-Danon et al. 91] Cerf-Danon H., El-Bèze M., *Three different probabilistic language models: Comparison and combination*, ICASSP'91, pp.297-300, Toronto (Canada), mai 1991.

[Chen et al. 98] Chen S., Beeferman D., Rosenfeld R., *Evaluation metrics for language models*, DARPA broadcast news transcription and understanding workshop, 1998.

[Chen et al. 99] Chen S., Goodman J., *An empirical study of smoothing techniques for language modeling*, Computer Speech and Language, 13:359-394, octobre 1999.

[Chibout et al. 99] Chibout K., Néel F., Mariani J., Masson N., *Ressources et évaluation en ingénierie des langues*, Actualité Scientifique, AUPELF-UREF, 1999.

[Clarkson et al. 97] Clarkson P., Rosenfeld R. *Statistical Language Modeling using the CMU-Cambridge toolkit*, Eurospeech'97, vol. 5, pp. 2707-2710, Rhodes (Grèce), septembre 1997.

[Combescure 81] Combescure P., *20 listes de dix phrases phonétiquement équilibrées*, Revue d'Acoustique n°56, pp 34-38, 1981.

[Cooley et al. 65] Cooley J.W., Tukey J.W., *An algorithm for the machine calculation of complex fourier series*, Mathematics of Computation, numéro 19, pp. 297-301, 1965.

[Damerau 64] Damerau F., *A technique for computer detection and correction of spelling errors*, Communications of the ACM, vol. 7, pp. 659-664, 1964.

[Davis et al. 80] Davis S., Mermelstein P., *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*, IEEE Transactions ASSP, ASSP-28(4), pp. 357-366, 1980.

[Deroo 98] Deroo O., *Modèles dépendant du contexte et méthodes de fusion de données appliqués à la reconnaissance de la parole par modèles hybrides HMM/MLP*, thèse de la faculté Polytechnique de Mons (Belgique), 196 pages, décembre 1998.

[Dolmazon et al. 97] Dolmazon J.M., Bimbot F., Adda G., El-Bèze M., Caërou J.C., Zeiliger J., Adda-Decker M., *Organisation de la première campagne aupelf pour l'évaluation des systèmes de dictée vocale*, actes des premières JST Francil, pp. 13-18. Avignon (France), 1997.

[Dutoit et al. 96] Dutoit T., Pagel V., Pierret N., Van Der Vreken O., Bataille F., *The MBROLA Project: Towards a Set of High-Quality Speech Synthesizers Free of Use for Non-Commercial Purposes*, ICSLP'96, Philadelphie (USA), 1996.

[El-Bèze et al. 90] El-Bèze M., Derouault A.M., *A morphological model for large vocabulary speech recognition*, ICASSP'90, pp. 577-580, 1990.

[Ferretti et al. 89] Ferretti M., Maltese G., Scarci S., *Language Model and Acoustic Model Information In Probabilistic Speech Recognition*, ICASSP'89, pp. 707-710, 1989.

[Fisher et al. 93] Fisher W.M., Fiscus J.G., *Better Alignment Procedures for Speech Recognition Evaluation*, ICASSP'93, volume 2, pp. 59-62, Albuquerque MN (USA), 1993.

[Forney 73] Forney G., *The Viterbi algorithm*, Proceedings of the IEEE (61), pp. 268-278, mars 1973.

[Gauvain et al. 00a] Gauvain J.L., Lamel L., Adda G., *Transcribing broadcast news for audio and video indexing*, Communications of the ACM, 43(2), pp. 64-70, février 2000.

[Gauvain 00b] Gauvain J.L., *Systèmes de reconnaissance à grand vocabulaire : progrès et défis*, JEP'2000, pp. 31-38, Aussois (France), juin 2000.

[Gildea et al. 99] Gildea D., Hoffman T., *Topic-Based Language Modeling Using EM*, Eurospeech'99, volume V, pp. 2167-2170 , Budapest (Hongrie), septembre 1999.

[Good 53] Good I.J., *The Population Frequencies of Species and the Estimation of Population Parameters*, Biometrika, volume 40, pp. 237-264, 1953.

[Goodman 01] Goodman J., *A bit of progress in language modeling, Extended Version*, Microsoft Research Technical Report MSR-TR-2001-72, disponible sur http://www.research.microsoft.com/~joshuago/.

[Gopalakrishnan et al. 95] Gopalakrishnan P., Bahl L., Mercer, R., *A tree search strategy for large vocabulary continuous speech recognition*, ICASSP'95, volume I, pp. 572-575, 1995.

[Haeb-Umback et al. 92] Haeb-Umback R., Geller D., Ney H., *Linear discriminant analysis for improved large vocabulary continuous speech recognition*, ICASSP'92, pp.13-16, volume I, 1992.

[Hermansky 90] Hermansky H. *Perceptual Linear Predictive (PLP) Analysis of speech*, Journal of Acoustic Society Am. 87(4), pp. 1738-1752, 1990.

[Hermansky et al. 94] Hermansky H., Morgan M., *RASTA processing of speech*, IEEE Transactions on Speech and Audio Processing 2(4), pp. 578-589, octobre 1994.

[Huang et al. 93] Huang X.D., Alleva F., Hon H.W., Hwang M.Y., Lee K.F., Rosenfeld R., *The Sphinx-II speech recognition system: An overview* Computer Speech and Language, 1993.

[Ito et al. 99] Ito A., Kohda M., Ostendorf M., *A New Metric For Stochastic Language Model Evaluation*, Eurospeech'99, volume IV, pp. 1591-1594 , Budapest (Hongrie), septembre 1999.

[Jardino 98] Jardino M., *Évaluation de modèles de langage à base de trigrammes de classes et de mots, avec le Jeu de Shannon*, XXIIèmes Journées d'Etudes sur la Parole, pages 363-366, Martigny (Suisse), 1998.

[Jelinek et al. 77] Jelinek F., Mercer R.L., Bahl L.R., Baker J.K., *Perplexity - A Measure of Difficulty of Speech Recognition Tasks*, 94th Meeting of the Acoustic Society of America, Miami Beach Floride, (USA), décembre 1977.

[Jelinek et al. 80] Jelinek F., Mercer R.L., *Interpolated estimation of Markov source parameters from sparse data*, Pattern Recognition in Practice, pp. 381-397, Amsterdam (Hollande), 1980.

[Jelinek 89] Jelinek F., *Self-Organized Language Modeling for Speech Recognition*, Readings in Speech Recognition, Alex Waibel and Kai-Fu Lee (Editors), in Morgan Kaufmann, 1989.

[Jelinek 01] Jelinek F., *Aspects of the Statistical Approach to Speech Recognition*, IEEE International Symposium on Information Theory, Washington D.C., juin 2001.

[Johnson et al. 99] Johnson S., Jourlin P., Moore G., Sparck Jones K., Woodland P., *The Cambridge University Spoken Document Retrieval*, ICASSP'99, papier numéro 2304, Phoenix Arizona (USA), mars 1999.

[Joux 98] Joux C., *Effets discursifs et calculs cognitifs sur les indices multimodaux mélodiques et oculaires en lecture*, DEA en psychologie cognitive de l'universtité Lumière de Lyon (Lyon 2), 1998.

[Katz 87] Katz S.M., *Estimation of probabilities from sparse data for the language model component of a speech recognizer*, IEEE Transactions on Acoustics, Speech and Signal Processing, volume ASSP-35, pages 400-401, mars 1987.

[Kenny et al. 91] Kenny P., Hollan R., Gupta V., Lennig M., Mermelstein P., O'Shaughnessy D., *A* admissible heuristics for rapid lexical access*, ICASSP'91, Volume I, pp. 689-693, Toronto (Canada), mai 1991.

[Klautau et al. 00] Klautau A., Jevtic N., Orlitsky A., *Server-Assisted Speech Recognition Over The Internet*, ICASSP'2000, Volume VI, Istanbul (Turquie), juin 2000.

[Kneser et al. 93a] Kneser R., Steinbiss V., *On the Dynamic Adaptation of Stochastic Language Models*, ICASSP'93, volume II, pp. 586-588, Minneapolis (USA), mai 1993.

[Kneser et al. 93b] Kneser R., Ney H., *Improved clustering techniques for classbased statistical language modelling*, Eurospeech'93, pp. 973-976, Berlin (Allemagne), septembre 1993.

[Kneser et al. 95] Kneser R., Ney H., *Improved backing-off for m-gram language modeling*, Conference on Acoustics, Speech and Signal Processing, volume 1, pages 181-184, 1995.

[Kuhn et al. 90] Kuhn R., De Mori R., *A Cache-Based Natural Language Model for Speech Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume PAMI-12, numéro 6, pp. 570-583, juin 1990.

[Langlois et al. 00] Langlois D., Smaïli K., Haton J.P., *Dealing with distant Relationships in Natural Language Modeling for Automatic Speech Recognition*, World Multiconference on Systemics, Cybernetics and Informatics, Orlando Florida (USA), juillet 2000.

[Lamel et al. 01] Lamel L., Gauvain J.L., Eskenazi M., *BREF, a large vocabulary spoken corpus for French*, Eurospeech'91, Gênes (Italie), 1991.

[Lamy 01] Lamy R., *Adaptation de modèles acoustiques et traitement des vecteurs acoustiques pour la reconnaissance automatique de la parole téléphonique*, DEA Informatique Système et Communication de Université Joseph Fourier de Grenoble, 2001.

[Lau 93] Lau R., *Maximum Likelihood Maximum Entropy Trigger Language Model*, thèse du Massachusetts Institute of Technology, mai 1993.

[Lazaridès et al. 98] Lazaridès A., Brousseau J., Lacouture R. Dumouchel P., *Le système de dictée vocale en français du CRIM utilisé pour l'ARC B1 de l'AUPELF*, exposé oral pour ARC-ILOR Thème B1, Reconnaissance de grands vocabulaire : dictée vocale, avril 1997.

[Levenshtein 66] Levenshtein V.I., *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Physics Doklady 10(8), pp. 707-710, février 1966.

[Levin et al. 98] Levin L., Gates D., Lavie A., Waibel A., *An Interlingua Based on Domaine Actions for Machine Translation of Task-Oriented Dialogues*, ICSLP'98, volume IV, pp. 1155-1158, Sydney (Australie), novembre-décembre 1998.

[McNair et al. 94] McNair A., Waibel A., *Improving Recognizer Acceptance through robust, natural Speech Repair*, ICSLP-94, volume IV, pp. 1299-1302, Yokohama (Japon), 1994.

[Miclet et al. 99] Miclet L., Chodorowski J., *Apprentissage et Évaluation de Modèles de Langage par des Techniques de Correction d'Erreurs*, TALN'99, pp. 253-262, Cargèse (Corse, France), 1999.

[Miller et al. 96] Miller J.W., Alleva F., *Evaluation of a language model using a clustered model backoff*, International Conference on Spoken Language Processing, pp. 390-393, 1996.

[Ney et al. 92] Ney H., Haeb-Umbach R., Tran B., Oerder M., *Improvements in Beam Search for 10.000 Word Continuous Speech Recognition*, ICASSP'92, volume I, pages 9-12, San Francisco (USA), mai 1992.

[Ney et al. 94] Ney H., Essen U. Kneser R, *On structuring probabilistic dependencies in stochastic language modeling*, Computer Speech and Language, volume 8(1), pp. 1-28, 1994.

[Perennou et al. 87] Pérennou G., De Calmès M., *BDLEX lexical data and knowledge base of spoken and written French*, European conference on Speech Technology, pp. 393-396, Edinburgh (Ecosse), septembre 1987.

[Ploux et al. 98] S. Ploux, B. Victorri, *Construction d'espaces sémantiques à l'aide de dictionnaires informatisés des synonymes*, T.A.L. volume 39, pp.161-182, 1998.

[Ponton 96] Ponton C., *Génération de textes en langue naturelle. Essai de définition d'un système noyau*, thèse de doctorat, Université Stendhal, Grenoble III, 1996.

[Rabiner et al. 89] Rabiner L. R., Juang B., *Tutorial on hidden Markov models and selected applications in speech recognition*, Proceedings of the IEEE, vol. 77, no. 2, pp. 257-285, 1989.

[Rosenfeld 94] Rosenfeld R., *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, thèse de l'université de Carnegie Mellon, Pittsburgh (USA), 114 pages, 1994.

[Rosenfeld 95] Rosenfeld R., *The CMU Statistical Language Modeling Toolkit, and its use*, ARPA Spoken Language Technology Workshop, Austin Texas (USA), pp. 47-50, 1995.

[Rosenfeld 00] Rosenfeld R., *Two decades of Statistical Language Modeling: Where Do We Go From Here?*, Proceedings of the IEEE, 88(8), 2000.

[Roussel 99] Roussel D., *Intégration de prédictions linguistiques issues d'applications à partir d'une grammaire d'arbres hors contexte. Contribution à l'analyse de la parole*, thèse de l'université Joseph Fourier de Grenoble, 401 pages, 1999.

[Sagayama 01] Sagayama S., *Analytic Methods for Acoustic Model Adaptation: a Review*, ITR-Workshop on Adaptation Methods For Speech Recognition, pp. 67-76, Nice Sophia-Antipolis (France), août 2001.

[Sakoe et al. 78] Sakoe H., Shiba S., *Dynamic programming algorithm optimization for spoken word recognition*, IEEE transaction on Acoustics, Speech and Signal Processing (26), pp. 143-165, 1978.

[Saon et al. 99] Saon G., Padmanablan M., *Data-driven Approach to Designing Compound Words for Continuous Speech Recognition*, ASRU'99 IEEE Workshop, pp. 261-265, Keystone Colorado (USA), décembre 1999.

[Savariaux et al. 97] Savariaux C., Farhat A., Héon M., O'Shaughnessy D., Lee C.Z., *Nouvelles avancées en reconnaissance de la parole continue grand vocabulaire du français basées sur le système de reconnaissance de l'INRS-télécommunications*, Actes des premières JST Francil 1997, pp. 31-34, Avignon (France), avril 1997.

[Schmid 94] Schmid H., *Probabilistic Part-of-Speech Tagging Using Decision Trees*, International Conference on New Methods in Language Processing, septembre 1994.

[Schukat-Talamazzini et al. 95] Schukat-Talamazzini E.G., Hendrych R., Kompe R., Niemann H., *Permugram language models*, Eurospeech'95, pp. 1773-1776. Madrid (Espagne), 1995.

[Sérasset et al 99] Sérasset G., Boitet C., *UNL-French deconversion as transfer & generation from an interlingua with possible quality enhancement through offline human interaction*, MT Summit 99, pp. 220-228, Singapour, septembre 1999.

[Shannon 48] Shannon C.E., *A Mathematical Theory of Communication*, Bell Systems Technical Journal, Volume 27, pp. 379-432 et 623-656, 1948.

[Shannon 51] Shannon C.E., *Prediction and Entropy of Printed English*, Bell Systems Technical Journal, Volume 30, pp. 50-64, 1951.

[Siohan 95] Siohan O., *On the robustness of linear discriminant analysis as preprocessing for noisy speech recognition*, ICASSP'95, pp. 125-128, volume I, Detroit (USA), 1995.

[Soong 91] Soong F.K., Huang E., *A tree-treillis bases fast search for finding the n best sentence hypotheses in continuous speech recognition*, ICASSP'91, pp. 705-708, volume I, Toronto (Canada), mai 1991.

[Spalanzani 99] Spalanzani A., *Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de reconnaissance automatique de la parole*, thèse de l'Université Joseph Fourier de Grenoble, 1999.

[Srihari et al. 92] Srihari R, Baltus C., *Combining statistical and syntactic methods in recognizing handwritten sentences*, AAAI Symposium : Probabilistic Approaches to Natural Language, pp. 121-127, 1992.

[Stevens et al. 40] Stevens S., Volkman J., *The relation of pitch to frequency*, American Journal of Psychology, vol. 53, 1940.

[Stork 99] Stork D.G., *The Hal 9000 Computer And The Vision Of 2001: A Space Odyssey*, Keynote Speech à ASRU'99 IEEE Workshop, Keystone Colorado (USA), décembre 1999.

[Taboada et al. 94] Taboada J., Feijoo S., Balsa R., Hernandez C., *Explicit estimation of speech boundaries*, IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159, 1994.

[Tanigaki et al. 00] Tanigaki K., Yamamoto H.., Sagisaka Y., *A hierarchical language model incorporating class-dependent word models for OOV recognition*, ICSLP'2000, volume III, pp. 123-126, 2000.

[Tomlison 91] Tomlison M.J., *Guide to Database Generation - Recording Protocol, Final Version*, SAM-RSRE-015, Marlvern, England.

[Umesh et al. 99] Umesh S., Cohen L., Nelson D., *Fitting the mel scale*, ICASSP'99, vol. 1, pp. 217-220, Phoenix Arizona (USA), mars 1999.

[Vaufreydaz et al. 98a] Vaufreydaz D., Akbar M., Caelen J., Serignat J.F., *EMACOP : Environnement Multimédia pour l'Acquisition et la gestion de COrpus Parole*, Journées d'Etude sur la Parole JEP'98, pp. 175-178, Martigny (Suisse), juin 1998.

[Vaufreydaz et al. 99a] Vaufreydaz D., Akbar M., Rouillard J., *A Network Architecture for Building Application that Use Speech Recognition and/or Synthesis*, Eurospeech'99, pp. 2159-2162, Budapest (Hongrie), septembre 1999.

[Vaufreydaz 99b] Vaufreydaz D., *Utilisation des documents en provenance d'Internet pour l'apprentissage de modèles de langage*, Rencontres Jeunes Chercheur en Parole (RJC'99), Avignon (France), novembre 1999.

[Vaufreydaz et al. 99c] Vaufreydaz D., Akbar M., Rouillard J., *Internet Documents : a Rich Source for Spoken Language Modelling*, ASRU'99 IEEE Workshop, pp. 277-281, Keystone Colorado (USA), décembre 1999.

[Vaufreydaz et al. 00] Vaufreydaz D., Bergamini C., Serignat J.F., Besacier L., Akbar M., *A New Methodology For Speech Corpora Definition From Internet Documents*, LREC'2000, pp. 423-426, Athènes (Grèce), juin 2000.

[Vaufreydaz et al. 01a] Vaufreydaz D., Besacier L., Bergamini C., Lamy R., *From generic to task-oriented speech recognition: French experience in the NESPOLE! European project*, ITR-Workshop on Adaptation Methods For Speech Recognition, pp. 179-182, Nice Sophia-Antipolis (France), août 2001.

[Vaufreydaz et al. 01b] Vaufreydaz D., Géry M., *Internet Evolution and Progress in Full Automatic French Language Modelling*, ASRU'01, 4 pages (CDROM), Trento (Italie), décembre 2001.

[Wagner et al. 74] Wagner R.A., Fisher M.J., *The string-to-string correction problem*, Journal of the Association for Computing Machinery, vol. 21, n°1, pp. 168-173, janvier 1974.

[Waibel et al. 91] Waibel A., Jain A., McNair A., Saito H., Hauptmann A., Tebelskis J., *JANUS : A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies*, ICASSP'91, volume II, pp. 793-796, Toronto (Canada), mai 1991.

[Witten et al. 91] Witten I.H., Bell T.C., *The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression*, IEEE Transactions Information Theory, vol. 34, numéro 4, pp. 1085-1094, 1991.

[Woszczyna 98] Woszczyna M., *Fast speaker independant large vocabulary continuous speech recognition*, thèse de l'université de Karlsruhe (Allemagne), 156 pages, février 1998.

[Zitouni 00] Zitouni I., *Modélisation du langage pour les systèmes de reconnaissance de la parole destinés aux grands vocabulaires*, thèse de l'université Henri Poincaré, Nancy I, 206 pages, mars 2000.

[Zhu et al. 01] Zhu X., Rosenfeld R., *Improving Trigram Language Modelling with the World Wide Web*, ICASSP'2001, Salt Lake City (USA), mai 2001.

Ouvrages

[Abramson 63] Abramson N., *Information Theory and Coding*, McGraw-Hill Book Company, USA, 1963.

[Bellanger 95] Bellanger M., *Traitement numérique du signal, Théorie et pratique*, éditions Masson, 1ère édition en 1980. Actuellement en 5ème édition, ISBN 2-225-84997-8, 1995.

[Black 00] Black U., *Voice Over IP*, édition Prentice Hall PTR, ISBN 0-13-022463-4, 2000.

[Blanche-Benveniste et al. 90] Blanche-Benveniste C., Bilger M., Rouget C., and Van Den Eynde K., *Le français parlé : études grammaticales*, CNRS Editions Paris France, ISBN 2-271-05535-0, 1990.

[Bristow 86] Bristow, G., *Electronic Speech Recognition: Techniques, Technology & Applications*, McGraw-Hill Book Company, USA, page xvii, 1986.

[Calliope 89] Calliope (groupe d'auteurs), *La parole et son traitement automatique*, Paris, MASSON, ISBN 2-225-81516-X, 1989.

[Grevisse 82] Grevisse M., *Le français correct - Guide pratique*, Duculot, Paris, actuellement en 3éme édition, ISBN 2-7242-1574-5, 1982.

[Habert et al. 92] Habert B., Nazarenko A., Salem A. *Les linguistiques de corpus*, Paris: Armand Colin - Masson, ISBN 2-200-01775-8, 1997.

[Jakobson 60] Jakobson R., *Linguistics and poetics*, Style in language, Seboek, Thomas, (ed.), MIT Press, pp. 350-377, 1960.

[Kerbrat 80] Kerbrat-Orecchioni C., *L'énonciation*, Paris : Armand Colin, page 19 de la 1ère édition, 1980. Actuellement en 4ème édition, ISBN 2-200-25019-3, 1999.

[Press et al. 92] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical recipes in C: The Art of Scientific Computing (seconde édition)*, édition Cambridge University Press, ISBN 0-521-43108-5, 1992. Téléchargeable gratuitement sur le site *http://www.ulib.org/webRoot/Books/Numerical_Recipes/*.

[Rabiner et al. 93] Rabiner L., Juang B.H., *Fundamentals of Speech Recognition*, édition Prentice Hall PTR, ISBN 0-130-15157-2, 1993.

[Salton et al. 83] Salton G., McGill M.J., *Introduction to Modern Information Retrieval*, McGraw-Hill Book Company, USA, ISBN 0-07-054484-0, 1983.

[Sankoff et al. 83] Sankoff D., Kruskal J.B., *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, ISBN 0-201-07809-0, 1983.

[Tannenbaum 96] Tannenbaum A., *Computer Networks 3rd Edition*, Prentice Hall PTR ed., ISBN 0-13-349945-6, 1996.

[Wall 00] Wall L., Christiansen T., Orwant J., *Programming Perl (Camel book), 3rd edition*, Edition O'Reilly, ISBN de la version française 2-84177-004-4, 2000.

[Walter 97] Walter H., *L'aventure des mots français venus d'ailleurs*, Éditions Robert Laffont, Paris, collection *Le livre de poche*, ISBN 2-253-14689-7, 1997.

[Walter 98] Walter H., *Le français d'ici, de là, de là-bas*, Éditions Robert Laffont, Paris, collection *Le Livre de Poche*, ISBN 2-253-14929-2, 1998.

[Welch 95] Welch B.B., *Practical Programming in Tcl and Tk*, Prentice Hall PTR, Upper Saddle River, New Jersey 07458. ISBN 0-13-182007-9, 1995.

Article de presse

[Presse 99] *La traduction orale automatique*, article de vulgarisation sur la démonstration du projet CSTAR-II paru dans la revue scientifique Sciences et Vie Micro (SVM) numéro 177 du mois de décembre 1999.

Documents de la Toile

[Web 01] *http://www.c-star.org/*, site du consortion CSTAR (Consortium for Speech Translation Advanced Research).

[Web 02] *http://nespole.itc.it/*, site du projet européen Nespole! (NEgotiating through SPOken Language in E-commerce).

[Web 03] *http://www.nist.gov/speech/*, site du groupe travaillant dans le domaine de la parole du National Institute of Standards and Technology (NIST).

[Web 04] *http://www.granddictionnaire.com/*, dictionnaire terminologique élaboré par l'Office de la Langue Française du Québec et par la société Semantix.

[Web 05] *http://alis.isoc.org/*, page d'accueil du projet Babel, un projet commun entre la société Alis Technologies et l'Internet Society portant sur l'internationalisation d'Internet.

[Web 06] *http://www.limsi.fr/TLP/grace/index.html*, pages décrivant l'action d'évaluation des analyseurs de textes nommée Grace de l'AUPELF.

[Web 07] *http://clips-index.imag.fr/*, site du notre robot de collecte de pages Web Clips-Index.

[Web 08] *http://abu.cnam.fr/*, site de l'Associations des Bibliophiles Universels.

[Web 09] *http://www.dcs.shef.ac.uk/~stu/com326/*, documents intitulé *Speech Recognition by Dynamic Time Warping* du département d'informatique de l'université de Sheffield.

[Web 10] *http://www.aethra.it/*, site de l'entreprise de télécommunication Aethra, partenaires du projet Nespole!.

[Web 11] *http://www.aupelf-uref.org/*, site de l'Agence Universitaire de la Francophonie (ex AUPELF-UREF).

[Web 12] *http://www.robotstxt.org/wc/robots.html*, site concentrant les informations concernant les robots de collecte de pages Web.

[Web 13] *http://www.icp.inpg.fr/Aupelf/B1/b1_spec1.html*, site décrivant l'ARC B1, concernant l'évaluation de système de dictée vocale, de l'AUPELF.

[Web 14] *http://www.icp.inpg.fr/ELRA/fr/index.html*, site de l'association ELRA (*European Language Resources Association*).

[Web 15] *http://www.nsrc.org/codes/country-codes.html*, site recensant les correspondances entre les codes des domaines de l'Internet et leurs significations.

Documents techniques

Les RFCs (*Request For Comment*) sont des documents techniques qui proposent la description de protocoles et de normes liés à Internet. Le lecteur pourra trouver ces documents un peu partout sur le réseau. Nous indiquons à titre d'exemple http://rfc.fh-koeln.de/, l'un des très nombreux sites regroupant ces documents.

[RFC 977] Request For Comment numéro 977 : *Network News Transfer Protocol, A Proposed Standard for the Stream-Based Transmission of News* proposé par le Network Working Group, 1986.

[RFC 1855] Request For Comment numéro 1855 : *Netiquette Guidelines* proposé par le Network Working Group, 1995.

[RFC 1866] Request For Comment numéro 1866 : *Hypertext Markup Language - 2.0* proposé par le Network Working Group, 1995.

[RFC 1945] Request For Comment numéro 1945 : *Hypertext Transfer Protocol -- HTTP/1.0* proposé par le Network Working Group, 1996.

[RFC 2616] Request For Comment numéro 2616 : *Hypertext Transfer Protocol -- HTTP/1.1* proposé par le Network Working Group, 1999.

[RFC 2980] Request For Comment numéro 2980 : *NNTP Common Extensions* proposé par le Network Working Group, 2000.

Publications personnelles

Publications scientifiques

[Besacier et al. 01a] Besacier L., Blanchon H., Fouquet Y., Guilbaud J.P., Helme S., Mazenot S., Moraru D., Vaufreydaz D., *Speech Translation for French in the NESPOLE! European Project*, Eurospeech'01, pp. 1291-1294, AAlborg (Danemark), septembre 2001.

[Besacier et al. 01b] Besacier L., Bergamini C., Vaufreydaz D., Castelli E., *The Effect of Speech and Audio Compression on Speech Recognition Performance*, Cannes (France), octobre 2001, à paraître.

[Besacier et al.] Besacier L., Vaufreydaz D., Grassi S., *Remote Recognition: the speech coding problem*, soumis à revue pour IEEE Transaction on Speech and Audio Processing.

[Meiye et al. 98] Meiye J.P., Rouillard J., Vaufreydaz D., *WebCompletion - protocole de normes associatives sur Internet*, Ecole Sémantique CNRS, CNRS, Asnelles-sur-mer (France), février 1998.

[Vaufreydaz et al. 98a] Vaufreydaz D., Akbar M., Caelen J., Serignat J.F., *EMACOP : Environnement Multimédia pour l'Acquisition et la gestion de COrpus Parole*, Journées d'Etude sur la Parole JEP'98, pp. 175-178, Martigny (Suisse), juin 1998.

[Vaufreydaz 98b] Vaufreydaz D., *EMACOP, un Environnement Multimédia pour l'Acquisition et la gestion de grands COrpus Parole*, rapport de DEA Informatique Système et Communications de l'UFR d'Informatique et de Mathématiques Appliquées, juin 1998.

[Vaufreydaz et al. 98c] Vaufreydaz D., Meiye J.P., Rouillard J., *WebCognition, expérimentations cognitives sur Internet*, 6ème Journée d'étude Créativité et Cognition, ACCION, Association de Jeunes Chercheurs en Science de la Cognition, pp. 12-13, Aix-en-Provence (France), novembre 1998.

[Vaufreydaz et al. 99a] Vaufreydaz D., Akbar M., Rouillard J., *A Network Architecture for Building Application that Use Speech Recognition and/or Synthesis*, Eurospeech'99, pp. 2159-2162, Budapest (Hongrie), septembre 1999.

[Vaufreydaz 99b] Vaufreydaz D., *Utilisation des documents en provenance d'Internet pour l'apprentissage de modèles de langage*, Rencontres Jeunes Chercheur en Parole (RJC'99), Avignon (France), novembre 1999.

[Vaufreydaz et al. 99c] Vaufreydaz D., Akbar M., Rouillard J., *Internet Documents : a Rich Source for Spoken Language Modelling*, ASRU'99 IEEE Workshop, pp. 277-281, Keystone Colorado (USA), décembre 1999.

[Vaufreydaz et al. 00] Vaufreydaz D., Bergamini C., Serignat J.F., Besacier L., Akbar M., *A New Methodology For Speech Corpora Definition From Internet Documents*, LREC'2000, pp. 423-426, Athènes (Grèce), juin 2000.

[Vaufreydaz et al. 01a] Vaufreydaz D., Besacier L., Bergamini C., Lamy R., *From generic to task-oriented speech recognition: French experience in the NESPOLE! European project*, ITR-Workshop on Adaptation Methods For Speech Recognition, pp. 179-182, Nice Sophia-Antipolis (France), août 2001.

[Vaufreydaz et al. 01b] Vaufreydaz D., Géry M., *Internet Evolution and Progress in Full Automatic French Language Modelling*, ASRU'01, 4 pages (CDROM), Trento (Italie), 2001.

Article de presse

[Presse 99] *La traduction orale automatique*, article de vulgarisation sur la démonstration du projet CStar-II paru dans la revue scientifique Sciences et Vie Micro (SVM) numéro 177 du mois de décembre 1999.

Annexes

Annexe A : choix des domaines Internet collectés

Présentation de l'annexe

Nous présentons ici les choix que nous avons faits pour les collectes de nos corpus en provenance de la Toile. Nous expliquerons nos motivations et donnerons finalement la liste complète des domaines que nous avons collectés. La correspondance entre les codes des domaines et leurs significations peut être trouvée sur [Web 15]. Ces codes sont définis par la norme ISO 3166-1.

Critères de sélection

Nous devions choisir un sous-ensemble de domaines de l'Internet pour nos collectes. En effet, vu la taille gigantesque de la Toile, nous ne pouvions pas envisager de tout collecter et d'ensuite tout filtrer par notre méthode des blocs minimaux. Nos choix sont motivés par l'optimisation de la taille des données françaises collectées, par des contraintes techniques et des besoins en recherche.

Notre premier critère concerne le volume des données françaises potentiellement présentes dans les corpus. Pour faire ce choix, nous avons utilisé deux paramètres. Le premier est le nombre de documents présents dans chacun des domaines. Nous obtenons cette information par le moteur de recherche *Altavista*. Le second est la part relative, dans chacun de ces domaines, des documents en français. Nous questionnons le moteur *AllTheWeb* pour cela. Le tableau VIII.4 nous donne un extrait des résultats pour quelques domaines.

Domaines	Nombre de pages	% de pages anglaises	% de pages françaises
be	1151946	31,62%	26,61%
ca	5228022	76,04%	20,29%
ch	2948797	25,60%	15,93%
com	113319060	83,21%	2,17%
de	17010456	18,03%	0,51%
edu	20744430	95,83%	0,23%
fr	3477169	19,98%	73,49%
gov	2308598	96,91%	0,11%
ma	39964	18,61%	76,22%
tn	17681	14,18%	66,98%

Tableau VIII.4 : Langues utilisées dans quelques domaines de la Toile

Les données du tableau précédent sont obsolètes à la date de la rédaction de ce manuscrit. Elles nous ont permis de choisir, lors de nos phases de collecte, les domaines que nous allions collecter. Dans l'éventualité de réaliser une nouvelle collecte, il faudrait refaire les estimations pour avoir une sélection de domaines représentative de l'état actuel de la Toile. Comme nous pouvons le voir, la taille, en nombre de pages, de certains domaines est très importante comme pour 'com', les domaines commerciaux. Cependant, la proportion de pages françaises est minime (2,17%). Il faudrait donc ramener une très grande quantité de données pour n'extraire finalement qu'une faible quantité d'information française. Nous avons fixé empiriquement un seuil pour faire un premier choix dans la liste des domaines. Nous avons pris 20%, qui paraissait être un bon compromis entre la taille de ce que nous collecterons et la part française que nous pourrions en extraire. Bien évidemment, le domaine 'fr', c'est-à-dire la France, correspond bien à ce critère.

Le second critère de filtrage était plus technique. Si l'on prend l'exemple du Canada (domaine 'ca'), il s'avère qu'il contient bien plus de 20% de pages françaises. Cependant, il est loin de nous, au sens du réseau, et donc de notre outils de collecte *Clips-Index*. Cela générait un très grand nombre d'erreurs d'attente sur des délais dépassés. Nous aurions pu accroître le temps d'attente mais cela au détriment de la vitesse de collecte. En effet, dans ce cas là, les processus passent plus de temps en attente qu'à ramener des données. Le bénéfice du parallélisme de notre outil de collecte s'amenuise alors. Nous avons supprimé les domaines posant problème.

Notre dernier choix fut à l'opposé du précédent. Nous voulions, pour des recherches en linguistiques ou en Recherche d'Informations, obtenir des données provenant de petits domaines francophones comme les pays du Maghreb ou encore le Vietnam. Même si ceux-ci pénalisent un peu la collecte car les réseaux de ces pays ne sont pas forcément, à l'heure actuelle, très performants, les données que nous y trouvons peuvent s'avérer très utiles pour faire des comparaisons entre les variations de l'emploi du français dans ces lieux. Ce choix s'est fait en collaboration avec l'équipe MRIM de notre laboratoire. Nous avons donc tenu compte de nos besoins et des leurs pour leurs travaux en Recherche d'Informations.

La section suivante de cette annexe nous donne, sous la forme d'un tableau, la liste complète des domaines que nous avons collectés.

Liste des domaines de la Toile collectés

Code de domaine	Correspondance	Code de domaine	Correspondance
ad	Principauté d'Andorre	ag	Antigua et Barbuda
be	Belgique	bf	Burkina Faso
bi	Burundi	bj	Bénin
bo	Bolivie	cd	République Démocratique du Congo
cf	République du Centre-Afrique	cg	Congo
ci	Côte d'Ivoire	ck	Îles Cook
cm	Cameroun	cu	Cuba
dj	Djibouti	dz	Danemark
eg	Égypte	fj	Fiji
fm	États Fédérés de Micronésie	fr	France
ga	Gabon	gd	Grenade
gf	Guyane Française	gn	Guinée
gp	Guadeloupe	gq	Guinée Équatoriale
int	International	jm	Jamaïque
jo	Jordan	kh	Cambodge

ky	Îles Caïmans	lb	Lebanon
lc	Sainte Lucie	li	Liechtenstein
ls	Lesotho	lu	Luxembourg
ma	Maroc	mc	Monaco
mg	Madagascar	ml	Mali
mq	Martinique	mr	Mauritanie
mu	Mauritius	nc	Nouvelle Calédonie
ne	Niger	ng	Nigeria
pf	Polynésie Française	qa	Qatar
re	la Réunion	rw	Rwanda
sc	Seychelles	sn	Sénégal
st	St. Tome	td	Tchad
tf	Territoire Français du Sud	tg	Togo
tn	Tunisie	tv	Tuvalu
va	Vatican	vn	Vietnam
vu	Vanuatu	wf	Wallis & Futuna
yt	Mayotte		

Tableau VIII.5 : Codes des domaines collectés et leurs significations (par ordre alphabétique)

Annexe B : liste des groupes de discussion de nos corpus

Présentation de l'annexe

Cette annexe présente la liste des groupes de discussion des corpus *NewFr1* et *NewsFr2*. *NewsFr1* comprend un ensemble de 234 groupes. *NewsFr2* contient 259 groupes. La différence entre les deux corpus correspond à des ajouts de nouveaux groupes et à la réorganisation de certains thèmes et sous-thèmes.

Liste des groupes de *NewFr1*

fr.announce.divers	fr.announce.important
fr.announce.seminaires	fr.bienvenue
fr.bienvenue.questions	fr.bio.biolmol
fr.bio.canauxioniques	fr.bio.general
fr.bio.genome	fr.bio.logiciel
fr.bio.medecine	fr.bio.medecine.veterinaire
fr.bio.pharmacie	fr.biz.d
fr.biz.produits	fr.biz.publicite
fr.biz.teletravail	fr.comp.applications.emacs
fr.comp.applications.groupware	fr.comp.applications.libres
fr.comp.applications.x11	fr.comp.developpement

fr.comp.divers	fr.comp.emulateurs
fr.comp.ia	fr.comp.infosystemes
fr.comp.infosystemes.www.annonces	fr.comp.infosystemes.www.annonces.d
fr.comp.infosystemes.www.auteurs	fr.comp.infosystemes.www.auteurs.php
fr.comp.infosystemes.www.divers	fr.comp.infosystemes.www.navigateurs
fr.comp.infosystemes.www.pages-perso	fr.comp.infosystemes.www.serveurs
fr.comp.lang.ada	fr.comp.lang.basic
fr.comp.lang.c	fr.comp.lang.c++
fr.comp.lang.general	fr.comp.lang.java
fr.comp.lang.lisp	fr.comp.lang.pascal
fr.comp.lang.perl	fr.comp.lang.tcl
fr.comp.mail	fr.comp.musique
fr.comp.objet	fr.comp.os.bsd
fr.comp.os.divers	fr.comp.os.linux.annonces
fr.comp.os.linux.configuration	fr.comp.os.linux.debats
fr.comp.os.linux.moderated	fr.comp.os.mac-os
fr.comp.os.ms-windows.programmation	fr.comp.os.ms-windows.win3
fr.comp.os.ms-windows.win95	fr.comp.os.ms-windows.winnt
fr.comp.os.msdos	fr.comp.os.os2
fr.comp.os.unix	fr.comp.os.unix.mac
fr.comp.os.vms	fr.comp.pao
fr.comp.peripheriques.modems	fr.comp.reseaux.ethernet
fr.comp.reseaux.ip	fr.comp.reseaux.supervision
fr.comp.securite	fr.comp.sys.amiga
fr.comp.sys.atari	fr.comp.sys.be
fr.comp.sys.divers	fr.comp.sys.mac
fr.comp.sys.mac.annonces	fr.comp.sys.mac.communication
fr.comp.sys.mac.materiel	fr.comp.sys.mac.programmation
fr.comp.sys.next	fr.comp.sys.palm-pilot
fr.comp.sys.parallele.sp.utilisateurs	fr.comp.sys.pc
fr.comp.sys.psion	fr.comp.text.tex
fr.doc.biblio	fr.doc.divers
fr.doc.magazines	fr.education.divers
fr.education.entraide	fr.education.entraide.maths
fr.education.medias	fr.education.superieur
fr.emplois.d	fr.emplois.demandes
fr.emplois.offres	fr.lettres.ecriture
fr.lettres.langue.anglaise	fr.lettres.langue.francaise

fr.misc.automoto.mecanique	fr.misc.bavardages.dinosaures
fr.misc.bavardages.linux	fr.misc.cryptologie
fr.misc.divers	fr.misc.droit
fr.misc.droit.internet	fr.misc.euro
fr.misc.finance	fr.misc.handicap
fr.misc.medias.presse-ecrite	fr.misc.transport.autostop
fr.misc.transport.rail	fr.misc.transport.urbain
fr.petites-annonces.divers	fr.petites-annonces.immobilier
fr.petites-annonces.informatique	fr.petites-annonces.informatique.logiciel
fr.petites-annonces.informatique.materiel	fr.petites-annonces.musique
fr.petites-annonces.vehicules	fr.rec.animaux
fr.rec.anime	fr.rec.apiculture
fr.rec.aquariophilie	fr.rec.arts.annonces
fr.rec.arts.bd	fr.rec.arts.litterature
fr.rec.arts.musique.autre	fr.rec.arts.musique.classique
fr.rec.arts.musique.hip-hop	fr.rec.arts.musique.jazz
fr.rec.arts.musique.pratique	fr.rec.arts.musique.rock
fr.rec.arts.plastiques	fr.rec.arts.polar
fr.rec.arts.sf	fr.rec.arts.spectacles
fr.rec.aviation	fr.rec.bateaux
fr.rec.bibliophilie	fr.rec.boissons.vins
fr.rec.bricolage	fr.rec.brocante
fr.rec.cinema.affiches	fr.rec.cinema.discussion
fr.rec.cinema.selection	fr.rec.cuisine
fr.rec.divers	fr.rec.genealogie
fr.rec.humour	fr.rec.jardinage
fr.rec.jeux.cartes	fr.rec.jeux.correspondance
fr.rec.jeux.divers	fr.rec.jeux.enigmes
fr.rec.jeux.jdr	fr.rec.jeux.jdr.par-forum
fr.rec.jeux.societe	fr.rec.jeux.video
fr.rec.jeux.video.materiel	fr.rec.jeux.video.tombraider
fr.rec.jeux.wargames	fr.rec.modelisme
fr.rec.montagne	fr.rec.moto
fr.rec.oracle	fr.rec.peche-chasse
fr.rec.philatelie	fr.rec.photo
fr.rec.plongee	fr.rec.radio
fr.rec.radio.amateur	fr.rec.son-image.home-cinema
fr.rec.sport.athletisme	fr.rec.sport.courir

fr.rec.sport.cyclisme	fr.rec.sport.divers
fr.rec.sport.equitation	fr.rec.sport.football
fr.rec.sport.roller	fr.rec.sport.rugby
fr.rec.sport.voile.planche	fr.rec.sport.vtt
fr.rec.tv.satellite	fr.rec.tv.series
fr.rec.tv.series.sf	fr.rec.voyages
fr.res-doct.archi	fr.reseaux.internet.fournisseurs
fr.reseaux.internet.hebergement	fr.reseaux.telecoms.adsl
fr.reseaux.telecoms.mobiles	fr.reseaux.telecoms.mobiles.bouygtel
fr.reseaux.telecoms.operateurs	fr.reseaux.telecoms.pabx
fr.reseaux.telecoms.rnis	fr.reseaux.telecoms.techniques
fr.sci.astronomie	fr.sci.automatique
fr.sci.biometrie	fr.sci.chimie
fr.sci.cogni.discussion	fr.sci.cogni.incognito
fr.sci.cogni.info	fr.sci.cogni.outil
fr.sci.cogni.publication	fr.sci.divers
fr.sci.electronique	fr.sci.geosciences
fr.sci.jargon	fr.sci.maths
fr.sci.philo	fr.sci.physique
fr.soc.alcoolisme	fr.soc.alternatives
fr.soc.divers	fr.soc.economie
fr.soc.histoire	fr.soc.homosexualite
fr.soc.internet	fr.soc.politique
fr.soc.religion	fr.soc.rural
fr.soc.sectes	fr.test
fr.usenet.8bits	fr.usenet.abus.d
fr.usenet.abus.rapports	fr.usenet.distribution
fr.usenet.divers	fr.usenet.forums.annonces
fr.usenet.forums.evolution	fr.usenet.logiciels
fr.usenet.reponses	fr.usenet.stats

Liste des groupes de *NewFr2*

fr.announce.divers	fr.announce.important
fr.announce.seminaires	fr.bienvenue
fr.bienvenue.questions	fr.bio.biolmol
fr.bio.canauxioniques	fr.bio.general

fr.bio.genome	fr.bio.logiciel
fr.bio.medecine	fr.bio.medecine.veterinaire
fr.bio.paramedical	fr.bio.pharmacie
fr.biz.d	fr.biz.produits
fr.biz.publicite	fr.biz.teletravail
fr.comp.applications.emacs	fr.comp.applications.genealogie
fr.comp.applications.groupware	fr.comp.applications.libres
fr.comp.applications.sgbd	fr.comp.applications.x11
fr.comp.developpement	fr.comp.divers
fr.comp.emulateurs	fr.comp.ia
fr.comp.infosystemes	fr.comp.infosystemes.www.annonces
fr.comp.infosystemes.www.annonces.d	fr.comp.infosystemes.www.auteurs
fr.comp.infosystemes.www.auteurs.php	fr.comp.infosystemes.www.divers
fr.comp.infosystemes.www.navigateurs	fr.comp.infosystemes.www.pages-perso
fr.comp.infosystemes.www.serveurs	fr.comp.lang.ada
fr.comp.lang.basic	fr.comp.lang.c
fr.comp.lang.c++	fr.comp.lang.general
fr.comp.lang.java	fr.comp.lang.lisp
fr.comp.lang.pascal	fr.comp.lang.perl
fr.comp.lang.tcl	fr.comp.mail
fr.comp.musique	fr.comp.objet
fr.comp.os.bsd	fr.comp.os.divers
fr.comp.os.linux.annonces	fr.comp.os.linux.configuration
fr.comp.os.linux.debats	fr.comp.os.linux.moderated
fr.comp.os.mac-os	fr.comp.os.mac-os.serveurs
fr.comp.os.ms-windows.programmation	fr.comp.os.ms-windows.win3
fr.comp.os.ms-windows.win95	fr.comp.os.ms-windows.winnt
fr.comp.os.msdos	fr.comp.os.os2
fr.comp.os.unix	fr.comp.os.unix.mac
fr.comp.os.vms	fr.comp.pao
fr.comp.peripheriques.modems	fr.comp.reseaux.ethernet
fr.comp.reseaux.ip	fr.comp.reseaux.supervision
fr.comp.securite	fr.comp.stockage
fr.comp.sys.amiga	fr.comp.sys.atari
fr.comp.sys.be	fr.comp.sys.calculatrices
fr.comp.sys.divers	fr.comp.sys.mac
fr.comp.sys.mac.annonces	fr.comp.sys.mac.communication
fr.comp.sys.mac.materiel	fr.comp.sys.mac.programmation

fr.comp.sys.next	fr.comp.sys.palm-pilot
fr.comp.sys.parallele.sp.utilisateurs	fr.comp.sys.pc
fr.comp.sys.psion	fr.comp.text.tex
fr.doc.biblio	fr.doc.divers
fr.doc.magazines	fr.education.divers
fr.education.entraide	fr.education.entraide.maths
fr.education.medias	fr.education.superieur
fr.emplois.d	fr.emplois.demandes
fr.emplois.offres	fr.lettres.ecriture
fr.lettres.langue.anglaise	fr.lettres.langue.francaise
fr.misc.automoto.mecanique	fr.misc.bavardages.dinosaures
fr.misc.bavardages.linux	fr.misc.cryptologie
fr.misc.depannage	fr.misc.divers
fr.misc.droit	fr.misc.droit.internet
fr.misc.euro	fr.misc.finance
fr.misc.handicap	fr.misc.medias.presse-ecrite
fr.misc.securite.routiere	fr.misc.transport.autostop
fr.misc.transport.rail	fr.misc.transport.urbain
fr.petites-annonces.divers	fr.petites-annonces.immobilier
fr.petites-annonces.informatique	fr.petites-annonces.informatique.logiciel
fr.petites-annonces.informatique.materiel	fr.petites-annonces.musique
fr.petites-annonces.vehicules	fr.rec.animaux
fr.rec.anime	fr.rec.apiculture
fr.rec.aquariophilie	fr.rec.aquariophilie.marine
fr.rec.arts.annonces	fr.rec.arts.bd
fr.rec.arts.litterature	fr.rec.arts.musique.autre
fr.rec.arts.musique.classique	fr.rec.arts.musique.hip-hop
fr.rec.arts.musique.jazz	fr.rec.arts.musique.pratique
fr.rec.arts.musique.rock	fr.rec.arts.plastiques
fr.rec.arts.polar	fr.rec.arts.sf
fr.rec.arts.sf.starwars	fr.rec.arts.spectacles
fr.rec.aviation	fr.rec.bateaux
fr.rec.bibliophilie	fr.rec.boissons.bieres
fr.rec.boissons.vins	fr.rec.bricolage
fr.rec.brocante	fr.rec.cinema.affiches
fr.rec.cinema.discussion	fr.rec.cinema.selection
fr.rec.cuisine	fr.rec.divers
fr.rec.genealogie	fr.rec.humour

fr.rec.jardinage	fr.rec.jardinage.bonsai
fr.rec.jeux.cartes	fr.rec.jeux.correspondance
fr.rec.jeux.divers	fr.rec.jeux.echecs
fr.rec.jeux.enigmes	fr.rec.jeux.jdr
fr.rec.jeux.jdr.par-forum	fr.rec.jeux.societe
fr.rec.jeux.video	fr.rec.jeux.video.materiel
fr.rec.jeux.video.tombraider	fr.rec.jeux.wargames
fr.rec.modelisme	fr.rec.montagne
fr.rec.moto	fr.rec.oracle
fr.rec.peche	fr.rec.peche-chasse
fr.rec.philatelie	fr.rec.photo
fr.rec.photo.labo	fr.rec.photo.materiel
fr.rec.photo.numerique	fr.rec.photo.pratique
fr.rec.plongee	fr.rec.radio
fr.rec.radio.amateur	fr.rec.son-image.home-cinema
fr.rec.sport.athletisme	fr.rec.sport.automobile
fr.rec.sport.courir	fr.rec.sport.cyclisme
fr.rec.sport.divers	fr.rec.sport.equitation
fr.rec.sport.football	fr.rec.sport.roller
fr.rec.sport.rugby	fr.rec.sport.voile.planche
fr.rec.sport.vtt	fr.rec.tv.satellite
fr.rec.tv.series	fr.rec.tv.series.sf
fr.rec.voyages	fr.res-doct.archi
fr.reseaux.internet.cable	fr.reseaux.internet.fournisseurs
fr.reseaux.internet.hebergement	fr.reseaux.telecoms.adsl
fr.reseaux.telecoms.mobiles	fr.reseaux.telecoms.mobiles.bouygtel
fr.reseaux.telecoms.operateurs	fr.reseaux.telecoms.pabx
fr.reseaux.telecoms.rnis	fr.reseaux.telecoms.techniques
fr.sci.astronomie	fr.sci.astronomie.amateur
fr.sci.astrophysique	fr.sci.automatique
fr.sci.biometrie	fr.sci.chimie
fr.sci.cogni.discussion	fr.sci.cogni.incognito
fr.sci.cogni.info	fr.sci.cogni.outil
fr.sci.cogni.publication	fr.sci.divers
fr.sci.electronique	fr.sci.electrotechnique
fr.sci.geosciences	fr.sci.jargon
fr.sci.maths	fr.sci.philo
fr.sci.physique	fr.soc.alcoolisme

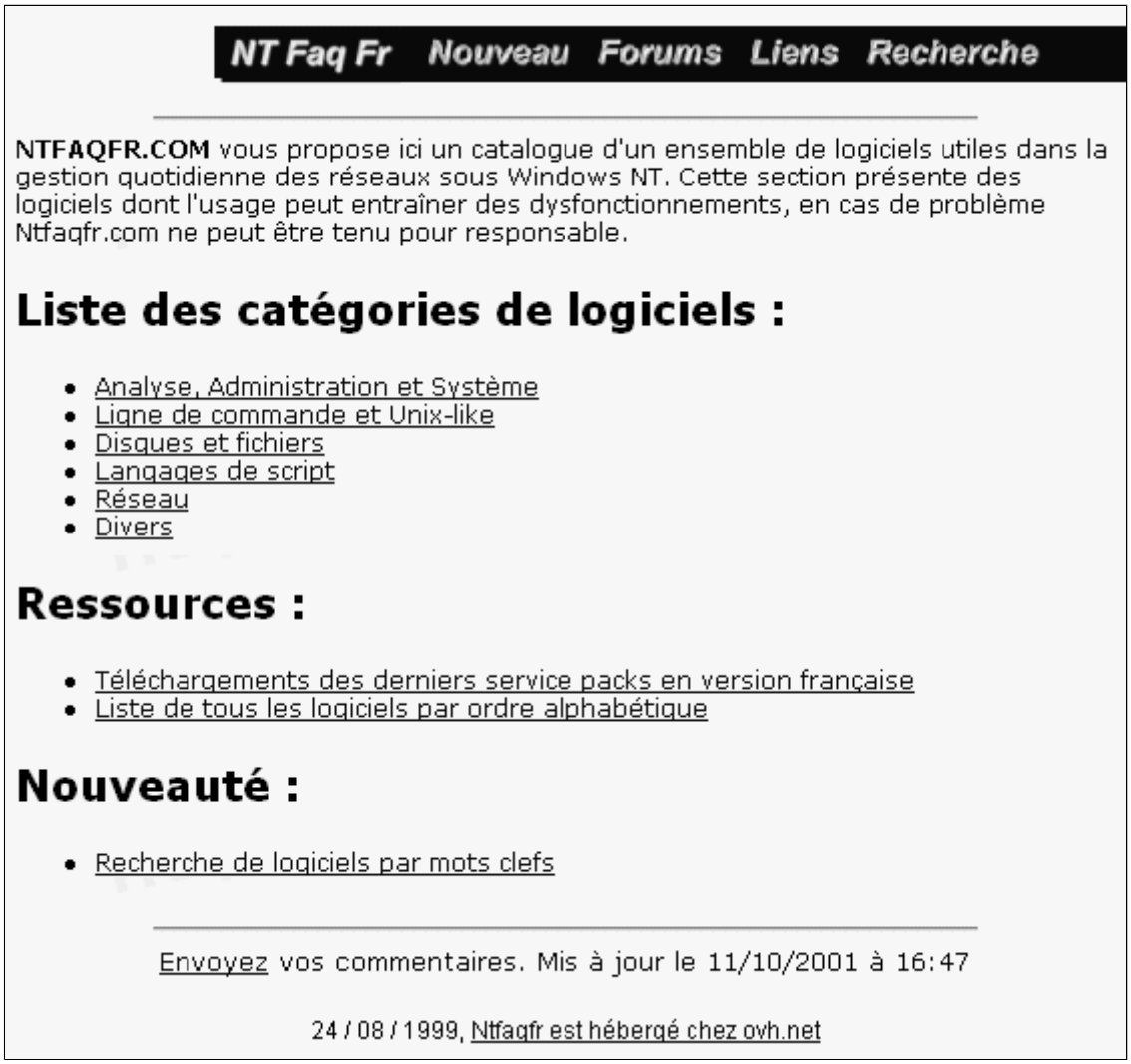
fr.soc.alternatives	fr.soc.divers
fr.soc.economie	fr.soc.environnement
fr.soc.histoire	fr.soc.homosexualite
fr.soc.internet	fr.soc.politique
fr.soc.religion	fr.soc.rural
fr.soc.sectes	fr.soc.travail
fr.test	fr.usenet.8bits
fr.usenet.abus.d	fr.usenet.abus.rapports
fr.usenet.distribution	fr.usenet.divers
fr.usenet.forums.annonces	fr.usenet.forums.evolution
fr.usenet.logiciels	fr.usenet.reponses
fr.usenet.stats	

Annexe C : exemple de page Web avec javascript

Présentation de l'annexe

Dans cette annexe, nous présentons un exemple de page Web dont nous sommes le Webmestre. Nous pouvons, sans problème de copyright, utiliser son contenu et afficher une représentation graphique de son rendu dans un navigateur. De par son domaine Internet, c'est-à-dire ".com", cette page ne fait pas partie de nos collections. Pourtant, elle illustre parfaitement le type de documents que nous trouvons par exemple dans *WebFr4*. Cette page fait partie de la FAQ (Foire Aux Questions) française du système d'exploitation Windows NT™ de Microsoft.

Rendu graphique de la page http://logiciels.ntfaqfr.com/



NT Faq Fr Nouveau Forums Liens Recherche

NTFAQFR.COM vous propose ici un catalogue d'un ensemble de logiciels utiles dans la gestion quotidienne des réseaux sous Windows NT. Cette section présente des logiciels dont l'usage peut entraîner des dysfonctionnements, en cas de problème Ntfaqfr.com ne peut être tenu pour responsable.

Liste des catégories de logiciels :

- Analyse, Administration et Système
- Ligne de commande et Unix-like
- Disques et fichiers
- Langages de script
- Réseau
- Divers

Ressources :

- Téléchargements des derniers service packs en version française
- Liste de tous les logiciels par ordre alphabétique

Nouveauté :

- Recherche de logiciels par mots clefs

Envoyez vos commentaires. Mis à jour le 11/10/2001 à 16:47

24 / 08 / 1999, Ntfaqfr est hébergé chez ovh.net

Code HTML de cette page

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Hide Java code from Java-disabled and Java-challenged browsers.
var isIMG = document.images; //See if the browser supports the images object
if (isIMG) {
ntfaqfron = new Image();
ntfaqfron.src = "images/faqfron.gif";
ntfaqfroff = new Image();
ntfaqfroff.src = "images/faqfroff.gif";
```

```
nouvon = new Image();
nouvon.src = "images/nouvon.gif";
nouvoff = new Image();
nouvoff.src = "images/nouvoff.gif";
forumson = new Image();
forumson.src = "images/forumson.gif";
forumsoff = new Image();
forumsoff.src = "images/forumsoff.gif";
lienson = new Image();
lienson.src = "images/lienson.gif";
liensoff = new Image();
liensoff.src = "images/liensoff.gif";
rechon = new Image();
rechon.src = "images/rechon.gif";
rechoff = new Image();
rechoff.src = "images/rechoff.gif";
espaceon = new Image();
espaceon.src = "images/espaceon.gif";
espaceoff = new Image();
espaceoff.src = "images/espaceoff.gif";
aideon = new Image();
aideon.src = "images/aideon.gif";
aideoff = new Image();
aideoff.src = "images/aideoff.gif";
}
function rollon(imgName) {
if (isIMG) {
imgOn = eval(imgName + "on.src");
document[imgName].src = imgOn;
}
}
function rolloff(imgName) {
if (isIMG) {
document[imgName].src = eval(imgName + "off.src");
}
}
// End of "hidden" Java code -->
</SCRIPT>
<STYLE TYPE="text/css">
<!--
BODY,TABLE {font-family : Verdana, Arial; font-size : 9pt}
H1 {font-family : Verdana, Arial; font-size : 16pt}
#rouge
{
text-decoration : none;
color : red;
}
#vert
{
text-decoration : none;
color : green;
```

```
}
#noir
{
text-decoration : none;
color : black;
}
#Description
{
color : black;
padding : 0.5em 0cm 0.5em 0;
text-decoration : none;
}

-->
</STYLE>
<TITLE>Les logiciels intéréssant pour NT</TITLE>
</HEAD>

<BODY BACKGROUND="images/backntfaq.gif"
BGCOLOR="#FFFFFF"
TEXT="#000000"
LINK="#000066"
VLINK="#666666">
<table border="0" width="100%" height="43">
<tr>
<TD ALIGN="LEFT">
<IMG SRC="http://perso.estat.com/cgi-bin/perso/21701758605?page=Accueil"
width="86" height="28">
</TD>
<td width="120%" height="39">
<p align="right"><a HREF="http://www.ntfaqfr.com/index.htm"
ONMOUSEOVER="rollon('ntfaqfr'); return false"
ONMOUSEOUT="rolloff('ntfaqfr'); return false"><img BORDER="0"
NAME="ntfaqfr"
ALT="NTFAQFr Départ" SRC="images/faqfron.gif" width="100"
height="40"></a><a
HREF="http://www.ntfaqfr.com/nouveau.html" ONMOUSEOVER="rollon('nouv');
return false"
ONMOUSEOUT="rolloff('nouv'); return false"><img BORDER="0" NAME="nouv"
ALT="News"
SRC="images/nouvoff.gif" width="90" height="40"></a><a
HREF="http://www.ntfaqfr.com/forums.html"
ONMOUSEOVER="rollon('forums'); return false" ONMOUSEOUT="rolloff
('forums'); return false"><img
BORDER="0" NAME="forums" ALT="Discussion Area"
SRC="images/forumsoff.gif" width="80"
height="40"></a><a HREF="http://www.ntfaqfr.com/liens.html"
ONMOUSEOVER="rollon('liens'); return false"
ONMOUSEOUT="rolloff('liens'); return false"><img BORDER="0" NAME="liens"
ALT="Links"
SRC="images/liensoff.gif" width="60" height="40"></a><a
```

```
HREF="http://www.ntfaqfr.com/recherche.html"
ONMOUSEOVER="rollon('rech'); return false" ONMOUSEOUT="rolloff('rech');
return false"><img
BORDER="0" NAME="rech" ALT="recherche" SRC="images/rechoff.gif"
width="90" height="40"></a><img
BORDER="0" NAME="espace" ALT="Space" SRC="images/espaceoff.gif"
width="70" height="40"><a
HREF="http://www.ntfaqfr.com/aide.html" ONMOUSEOVER="rollon('aide'); return
false"
ONMOUSEOUT="rolloff('aide'); return false"><img BORDER="0" NAME="aide"
ALT="Help"
SRC="images/aideoff.gif" width="62" height="40"></a> </td>
</tr>
</table>
<hr WIDTH="75%" align="center">
<STRONG>NTFAQFR.COM</STRONG> vous propose ici un catalogue d'un
ensemble de logiciels
utiles dans la gestion quotidienne des réseaux sous Windows NT.
Cette section présente des logiciels dont l'usage peut entraîner des
dysfonctionnements, en cas de problème Ntfaqfr.com ne peut être tenu pour
responsable.
<H1>Liste des catégories de logiciels :</H1>
<UL>
<LI><A HREF="AnalyseAdminEtSysteme.html">Analyse, Administration et
Système</A> </LI>
<LI><A HREF="CommandeEtUnixLike.html">Ligne de commande et Unix-like</A>
</LI>
<LI><A HREF="Disque.html">Disques et fichiers</A> </LI>
<LI><A HREF="LangagesDeScript.html">Langages de script</A> </LI>
<LI><A HREF="Reseau.html">Réseau</A></LI>
<LI><A HREF="Divers.html">Divers</A> </LI>
</UL>
<H1>Ressources :</H1>
<UL>
<LI><A HREF="Telechargements.html">Téléchargements des derniers service packs
en version française</A></LI>
<LI><A HREF="Abc.html">Liste de tous les logiciels par ordre
alphabétique</A></LI>
</UL>
<H1>Nouveauté :</H1>
<UL>
<LI><A HREF="Recherche.html">Recherche de logiciels par mots clefs</A></LI>
</UL>
<HR WIDTH="75%" ALIGN="CENTER">
<CENTER>
<FONT SIZE="2">
<A HREF="mailto:logiciels@ntfaqfr.com">Envoyez</A> vos commentaires.
<SCRIPT LANGUAGE="JavaScript">
<!--
majd = new Date (document.lastModified)
document.write("Mis à jour le ",majd.getDate(),"/", majd.getMonth()+1,"/",
```

```
majd.getFullYear()," à ",majd.getHours(),":");
if ( majd.getMinutes() < 10 )
document.write("0");
document.write( majd.getMinutes() );
// End of "hidden" Java code -->
</SCRIPT>
</FONT>
</CENTER>
<P ALIGN="CENTER"><FONT FACE="Arial">
24 / 08 / 1999, <A HREF="http://www.ovh.net" TARGET="_top">Ntfaqfr est
hébergé chez ovh.net </A>
<SMALL><BR></SMALL></FONT><BR>
</BODY>
</HTML>
```

Texte extrait par notre filtre Html2Text

NTFAQFR . COM vous propose ici un catalogue d' un ensemble de logiciels utiles dans la gestion quotidienne des réseaux sous Windows NT . Cette section présente des logiciels dont l' usage peut entraîner des dysfonctionnements , en cas de problème Ntfaqfr . com ne peut être tenu pour responsable . Liste des catégories de logiciels : Analyse, Administration et Système . Ligne de commande et Unix-like . Disques et fichiers . Langages de script . Réseau . Divers . Ressources : Téléchargements des derniers service packs en version française . Liste de tous les logiciels par ordre alphabétique . Nouveauté : Recherche de logiciels par mots clefs . Envoyez vos commentaires . Mis à jour le onze / dix / deux mille un à dix-huit heures cinquante vingt quatre / zéro huit / mille neuf cent quatre-vint dix-neuf , Ntfaqfr est hébergé chez ovh . net .

Annexe D : balises HTML et leurs équivalents

Présentation de l'annexe

Nous trouverons dans cette annexe, la liste des balises ("tags") de la norme HTML [RFC 1866]. Cette liste est en fait le fichier de configuration de notre outil d'extraction de texte à partir de pages Web. Sur chaque ligne est présenté un marqueur avec une indication du traitement que le filtre doit réaliser. Si le marqueur est suivi d'un signe moins, alors aucune information ne vient remplacer ce marqueur. Si il y a un signe plus, alors le marqueur HTML est remplacé par un espace. Enfin si c'est un point, le marqueur est remplacé par un point car il représente la fin possible d'une phrase. C'est le cas des cellules d'un tableau par exemple. Cela nous permet d'ajouter de l'information basique extraite de la structure du document HTML.

Ce fichier de configuration permet d'ajouter de nouveaux éléments, si la norme HTML est étendue dans le futur ou pour traiter d'autres données, par exemple au format SGML.

Fichier de configuration du programme Html2Text

```
# Variable définies dans HTML
# - TAG ne generant pas d'espace
# + TAG generant un espace
# . TAG generant un pseudo . et un espace
<!DOCTYPE -
<A -
</A> -
<ADDRESS +
</ADDRESS> .
<APPLET -
```

</APPLET> -
<AREA -
<B -
</B> -
<BASE -
<BASEFONT -
<BGSOUND -
<BIG -
</BIG> -
<BLINK -
</BLINK> -
<BLOCKQUOTE +
</BLOCKQUOTE> .
<BODY +
</BODY> -
<BR +
<CAPTION +
</CAPTION> .
<CENTER +
</CENTER> .
<CITE -
</CITE> -
<CODE -
</CODE> -
<COL +
</COL> .
<COLGROUP -
<COMMENT -
<CSOBJ -
</CSOBJ> -
<DD +
</DD> +
<DFN -
</DFN> -
<DIR +
</DIR> .
<DIV +
</DIV> .
<DL +
</DL> .
<DT .
</DT> .
<EM -
</EM> -
<EMBED +
<FONT -
</FONT> -
<FORM +
</FORM> +
<FRAME +
<FRAMESET +

</FRAMESET> +
<H1 +
</H1> .
<H2 +
</H2> .
<H3 +
</H3> .
<H4 +
</H4> .
<H5 +
</H5> .
<H6 +
</H6> .
<HEAD +
</HEAD> -
<HR .
<HTML -
</HTML> -
<I -
</I> -
<IFRAME -
</IFRAME> -
<IMG +
<INPUT +
<ISINDEX +
<KBD -
</KBD> -
<LI +
</LI> .
<LINK -
</LINK> -
<LISTING +
</LISTING> +
<MAP +
</MAP> +
<MARQUEE -
</MARQUEE> -
<MENU -
</MENU> +
<META -
<MULTICOL -
</MULTICOL> .
<NEXTID -
<NOBR -
</NOBR> -
<NOFRAMES -
</NOFRAMES> -
<NOSCRIPT -
</NOSCRIPT> -
<OBJECT +
</OBJECT> +

<OL +
</OL> .
<OPTION -
<P -
</P> .
<PLAINTEXT -
</PLAINTEXT> -
<PRE -
</PRE> -
<S -
</S> -
<SAMP -
</SAMP> -
<SCRIPT -
</SCRIPT> -
<SELECT -
</SELECT> -
<SMALL -
</SMALL> -
<SOUND -
<STRIKE -
</STRIKE> -
<STRONG -
</STRONG> -
<SPACER +
<SPAN -
</SPAN> -
<STYLE -
</STYLE> -
<SUB -
</SUB> -
<SUP -
</SUP> -
<TABLE +
</TABLE> +
<TBODY -
</TBODY> -
<TD +
</TD> .
<TEXTAREA -
</TEXTAREA> .
<TFOOT -
</TFOOT> -
<TH -
</TH> .
<THEAD -
</THEAD> .
<TITLE -
</TITLE> .
<TR -
</TR> .

<TT -
</TT> .
<U -
</U> .
<UL -
</UL> .
<VAR -
</VAR> -
<WBR +
<XMP -
</XMP> -

Annexe E : exemple d'un message extrait d'un *newsgroup*

Présentation de l'annexe

Nous trouvons dans cette annexe un exemple de message publié dans les forums de discussion par nous même. Il est une réponse à une question posée dans le newsgroup fr.comp.os.ms-windows.winnt. Dans un premir temps, nous donnons le message complet tel qu'il est en envoyé à notre robot News-Index par le serveur de *newsgroups*. Nous trouvons ensuite le texte que nous en avons extrait.

Message original complet

```
From: "Dominique Vaufreydaz"
Newsgroups: fr.comp.os.ms-windows.winnt
Subject: Re: Macfile : probleme de fou
Date: Tue, 2 Oct 2001 15:06:02 +0200
Organization: IMAG
Lines: 11
Message-ID: <9pce3r$6oj$1@trompette.imag.fr>
References:
NNTP-Posting-Host: ares.imag.fr
X-Trace: trompette.imag.fr 1002027963 6931 129.88.41.79 (2 Oct 2001 13:06:03
GMT)
X-Complaints-To: abuse@imag.fr
NNTP-Posting-Date: 2 Oct 2001 13:06:03 GMT
X-Priority: 3
X-MSMail-Priority: Normal
X-Newsreader: Microsoft Outlook Express 6.00.2462.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2462.0000
```

Bonjour,

"XX XX" wrote in message news:FTiu7.956$fy6.1059865@nnrp5.proxad.net...
> A partir d'un Windows 95, tantôt on peut renommer les fichiers, tantôt on a
> le message "Violation de partage"...

Probleme avec un antivirus ???

Doms.

Texte extrait

re : macfile : probleme de fou . bonjour , " xx xx " < xx . xx @free . fr > wrote in message news : ftiu7 . 956$fy6 . 1059865 @nnrp5 . proxad . net . . . a partir d' un windows 95 , tantôt on peut renommer les fichiers , tantôt on a le message " violation de partage " . . . probleme avec un antivirus . . . doms .

Annexe F : liste des mots composés

Présentation de l'annexe

Cette annexe contient la liste des mots composés que nous avons extraits automatiquement pour la tâche de réservation touristique à partir de *WebFr*.

Liste des mots composés de la tâche de réservation touristique

a-t-il	a-t-on
assez d'	au-delà du
au-dessous du	au-dessus du
au bout d'	au courant d'
au cours d'	au début des
au jour d'aujourd'hui	au lieu d'
au lieu de	au niveau d'
au niveau des	au niveau du
autour d'	beaucoup d'
c'est-à-dire qu'	ce qu'
comment est-ce qu'	d'aujourd'hui
dans ces conditions	dans l'ensemble
dans la mesure où	dans le cas où
dans le temps	de l'
de plus en plus	de temps en temps
de toute façon	est-ce qu'
est-ce que	il y a
la plupart des	loin d'
lors d'	metteur en scène
mises en scènes	n'est-ce pas
n'importe comment	n'importe lequel
n'importe quel	n'importe quelle
n'importe qui	n'importe quoi
nec plus ultra	niveau de vie

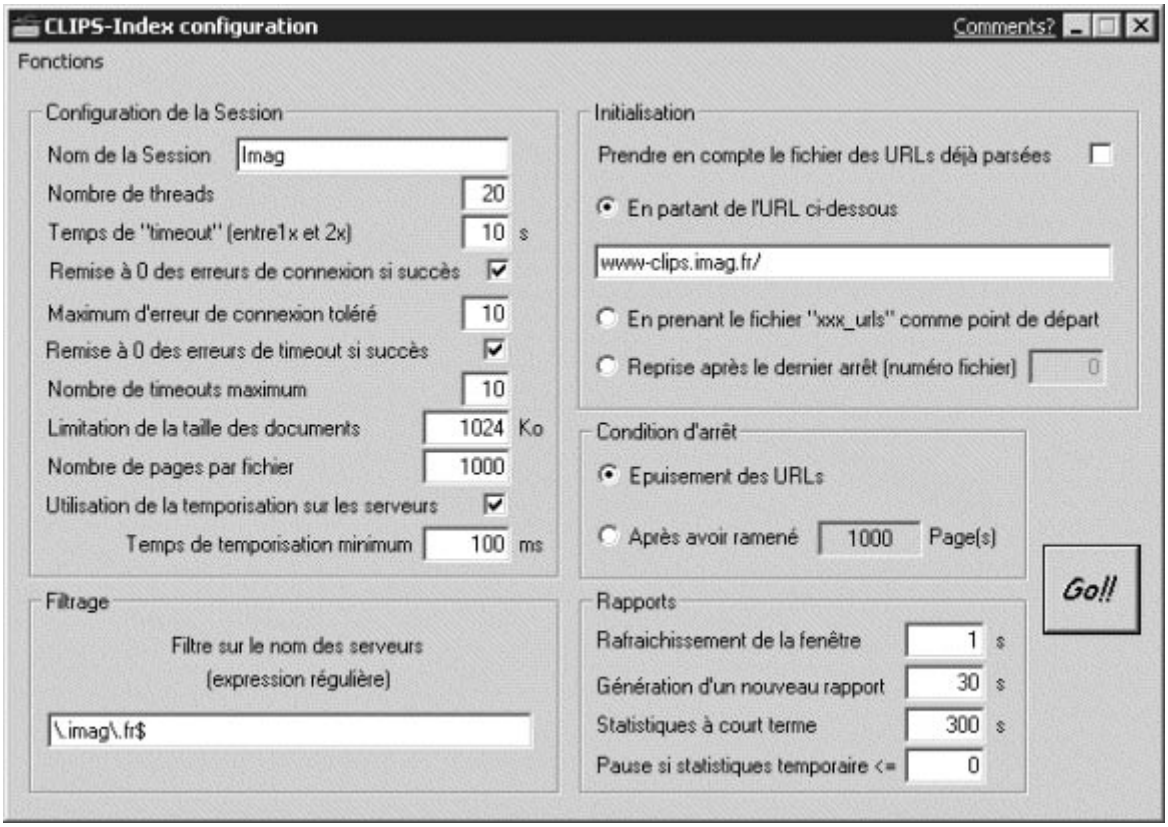
où est-ce qu'	par la suite
par rapport au	par rapport aux
par rapport à	plein d'
plus d'	plus d'un
plus d'une	plus du tout
plus ou moins	point de vue
points de vue	pour l'instant
pour le moment	prix de revient
qu'est-ce	qu'est-ce qu'
qu'est-ce que	qu'est-ce qui
quand est-ce qu'	quatre-vingt-cinq
quatre-vingt-deux	quatre-vingt-dix
quatre-vingt-dix-huit	quatre-vingt-dix-neuf
quatre-vingt-dix-sept	quatre-vingt-douze
quatre-vingt-huit	quatre-vingt-neuf
quatre-vingt-onze	quatre-vingt-quatorze
quatre-vingt-quatre	quatre-vingt-quinze
quatre-vingt-seize	quatre-vingt-sept
quatre-vingt-six	quatre-vingt-treize
quatre-vingt-trois	quatre-vingt-un
roulement à billes	s'il vous plaît
sans aucun doute	sans qu'
si bien qu'	si bien que
soixante-dix-huit	soixante-dix-neuf
soixante-dix-sept	sorte d'
suivant qu'	tandis qu'
tant qu'	tout au moins
tout compte fait	tout de même
tout de suite	tout le monde
tout le temps	tout à fait
traveller's chèque	une fois que
vingt et un	ça qu'
étant donné qu'	

Annexe G : captures d'écran de Clips-Index

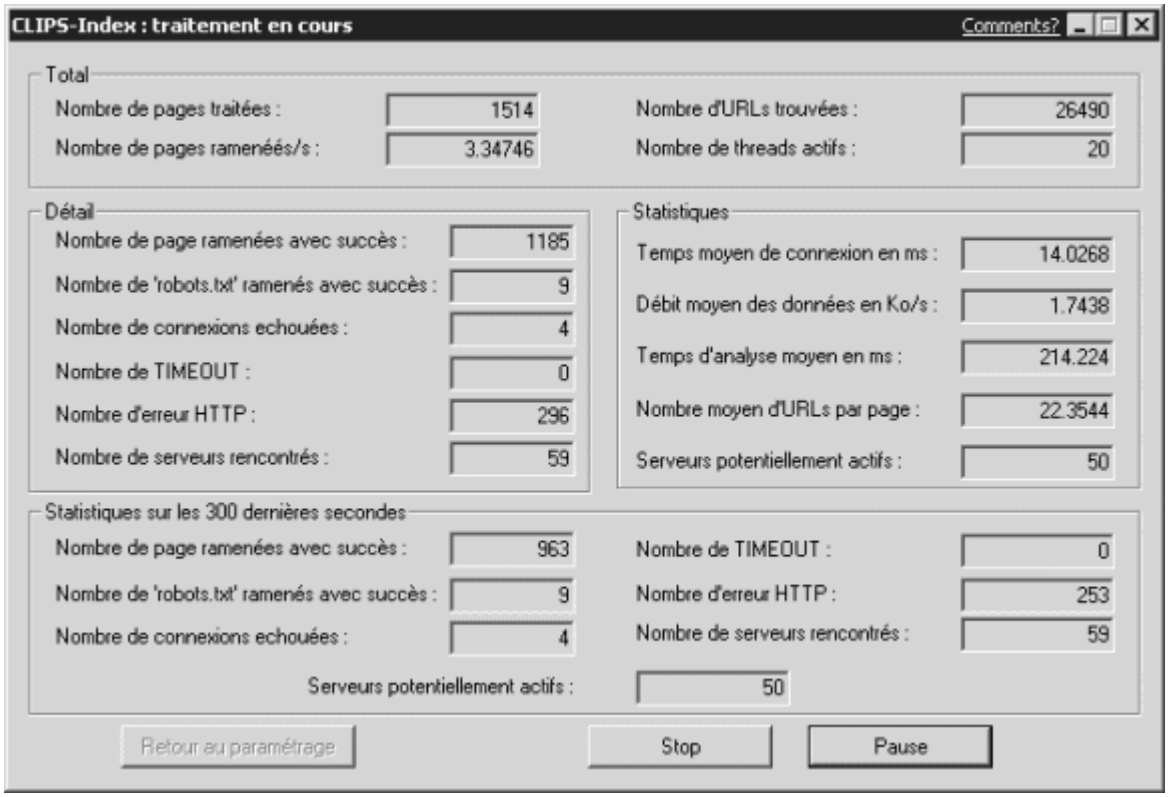
Présentation de l'annexe

Nous trouvons dans cette annexe deux captures d'écrans du logiciel Clips-Index. La première nous montre les paramètres qu'il est possible de règler. La seconde correspond à l'affiche de Clips-Index pendant un phase de collecte de documents de la Toile. Il est ainsi possible d'avoir des statistiques à long et à cours terme du travail du Robot.

Capture de l'interface de configuration de Clips-Index



Capture de l'interface de travail de Clips-Index



Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue

Résumé : Les ressources textuelles sont celles qui font le plus défaut dans les recherches sur la modélisation statistique du langage, surtout pour l'apprentissage de modèles adaptés au dialogue. Cette thèse propose d'utiliser les documents en provenance d'Internet pour l'apprentissage de tels modèles. La collecte de plusieurs corpus a permis la mise en évidence de certaines propriétés intéressantes. Celles-ci concernent la quantité de texte, le nombre de vocables différents mais surtout la possibilité de trouver des formes propres à l'expression orale. Ces formes ne se trouvent pas dans les corpus journalistiques qui sont pourtant très largement employés. L'évolution de ces documents au cours des dernières années a encore accru cette adéquation. La thèse introduit alors une nouvelle méthode, entièrement automatique, de calcul de modèles de langage à partir de ces données. Elle commence par un filtrage dit par " blocs minimaux " basé sur le vocabulaire de l'application visée. Ensuite, le calcul du modèle de langage statistique, type n-gramme, se fait au prix d'une légère adaptation des algorithmes standards dans le domaine. Les résultats de cette méthode sont de l'ordre de 90% de taux de reconnaissance pour des petits vocabulaires et de 80% pour de plus larges vocabulaires. De plus, les résultats obtenus, sans aucune adaptation, sur une base sonore état de l'art de l'AUPELF sont du même ordre que ceux des autres laboratoires ayant participé à l'évaluation. La thèse présente aussi d'autres applications d'Internet. Ainsi, L'utilisation de la hiérarchie des newsgroups permet la mise au point d'un détecteur de thème fondé sur une normalisation de modèles unigrammes. Ses performances sont d'environ 70%. L'intégration de ce détecteur au sein des algorithmes de reconnaissance de la parole permet un gain de 5% en taux de reconnaissance. Enfin, une adaptation de la méthode des blocs minimaux a été utilisée pour faciliter la définition d'un ensemble de phrases pour l'enregistrement d'un corpus sonore.

Mots-clefs : Modélisation statistique du langage, reconnaissance de la parole, Internet, détection de thèmes.

Statistical language modelling using Internet documents for continuous speech recognition

Summary : In statistical language modelling researches, there is a lack of huge text corpora, especially for spoken language modelling. This thesis deals with using Internet documents in order to train such statistical models. After gathering corpora, we highlighted several interesting properties like the huge quantity of text, the number of different French lexical forms and especially the ability of finding spoken dialog utterances. This kind of utterances is not present in usual journalistic corpora even if these corpora are widely used. During the past years, the evolution of Internet documents increased this adequacy. This thesis also introduces a new fully automatic method to compute statistical language models on Internet data. This method starts with a special filter called "minimal blocks" only based on the lexicon. Next, with modified computing algorithms, we can obtain statistical models like n-grams. Results using this method are about 90% of word accuracy for small vocabulary and about 80% of words accuracy for larger ones. Moreover, results on a state of the art audio corpus given by AUPELF for evaluation, without any kind of adaptation, are close to those obtained by other research teams. In this thesis, we also report other applications of Internet documents. Indeed, using the French newsgroups hierarchy, we can compute a topic detector based on normalized unigrams models. Topic detection accuracy is about 70%. Using this topic detector in speech recognition algorithms can increase word accuracy by up to of 5%. At last, a derived approach from "minimal blocks" method has been applied to define a set of sentences to record an audio corpus.

Keywords: Statistical language modelling, continuous speech recognition, Internet, Topic detection.

Laboratoire CLIPS, Université Joseph Fourier et Fédération IMAG, BP 53, 38041 Grenoble Cedex 9